

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Aleksander Bešir

**Razvoj elektronske knjige gostov z
urejevalnikom rezervacij za manjše
nastanitvene objekte**

DIPLOMSKO DELO
UNIVERZITETNI ŠTUDIJ RAČUNALNIŠTVO IN
INFORMATIKA

MENTOR: prof. dr. Franc Solina

Ljubljana, 2015

Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Preučite problematiko vodenja elektronske knjige gostov in urejanja rezervacij v manjših nastavitvenih objektih. Na osnovi analize zahtev in analize obstoječih programskih rešitev naredite načrt in implementirajte programsko opremo, ki bo izpolnjevala zadane zahteve. Za programsko opremo, razvito v okviru diplomske naloge in ki bo uporabnikom na voljo brezplačno, izberite najbolj primeren model za distribucijo.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Aleksander Bešir sem avtor diplomskega dela z naslovom:

Razvoj elektronske knjige gostov z urejevalnikom rezervacij za manjše nastanitvene objekte.

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom prof. dr. Franca Soline,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 19. januarja 2015

Podpis avtorja:

Mentorju prof. dr. Francu Solini se zahvaljujem za uso pomoč, vodenje in podporo, ki so mi bili na voljo tako med pisanjem naloge kot v celotnem času študija. Lektorici Petri Mlakar gre zahvala za končni pregled in ureditev besedila. Sorodnikom se iskreno zahvaljujem za razumevanje, mir in predvsem udobje, ki so mi ga nudili med implementacijo lastne rešitve. Nazadnje pa bi se rad še posebej zahvalil Karin Velikonja za motivacijo, brez katere bi bila izdelava naloge veliko težja.

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Potrebe manjših nastanitev	7
2.1	Administrativno recepcijsko delo brez namenske računalniške opreme	9
2.1.1	Zapis rezervacij	10
2.1.2	Urejanje rezervacij	14
2.1.3	Prijava gostov	15
2.1.4	Vodenje statistike in priprava poročil	18
2.1.5	Upravljanje kanalov	21
2.1.6	Priprava računov	22
2.2	Digitalizacija recepcijskega dela	25
2.2.1	Stopnja sklopljenosti modulov	26
2.2.2	Preprečevanje napak	28
2.2.3	Število sočasnih uporabnikov	30
2.2.4	Hranjenje podatkov	31
2.2.5	Vzdrževanje programske opreme in podpora	32
3	Pregled obstoječih rešitev na slovenskem trgu	35
3.1	Recepcija Polis	36

KAZALO

3.2	iGost	41
3.3	ProGOST	45
3.4	Birokrat Hotelir	46
3.5	Pomanjkljivosti predstavljenih rešitev	47
4	Lastna rešitev	49
4.1	Zasnova programa in načrt razvoja	49
4.1.1	Iterativno inkrementalni in uporabniško usmerjen razvoj	50
4.1.2	Trislojna arhitektura	53
4.1.3	Manj modulov, več funkcij	55
4.2	Programski paket eGost	56
4.2.1	Delovanje in integracija	57
4.2.2	Predpisana oblika prijavnice	59
4.3	Implementacija	61
4.3.1	Platforma in orodja	61
4.3.2	Podatkovni sloj	62
4.3.3	Logični sloj	65
4.3.4	Prezentacijski sloj	66
4.4	Moduli in funkcionalnosti	71
4.4.1	Knjiga gostov	74
4.4.2	Urejevalnik rezervacij	86
4.4.3	Dodatna orodja	98
4.4.4	Namestitveni program	99
4.4.5	Spletna stran s forumom	101
5	Odziv prvih uporabnikov	103
5.1	Analitika	104
5.2	Tržni potencial in poslovni model	105
5.3	Nadaljnji razvoj	107
6	Sklepne ugotovitve	109

Povzetek

V diplomskem delu so opisani motivacija za razvoj, načrt in delovanje programske opreme, ki implementira elektronsko knjigo gostov z urejevalnikom rezervacij za majhne in srednje velike nastanitvene objekte. Prvi del zajema predstavitev ciljne skupine uporabnikov, opis njihovega dela brez programske opreme, pregled njihovih potreb, analizo obstoječih rešitev na slovenskem trgu in identifikacijo njihovih pomanjkljivosti, zaradi katerih obstaja potreba po novi rešitvi. Sledijo opisi zasnove lastne rešitve, ki poskuša to potrebo zadovoljiti, načrta njenega razvoja in predstavitev programskega paketa eGost, ki omogoča varno pošiljanje podatkov o gostih preko spleta. Predstavljena je implementacija rešitve v obliki brezplačnega programa Opengost, funkcije, ki jih ta omogoča, pa so podrobno opisane. V zaključnem delu je predstavljen odziv prvih uporabnikov, opisana pa je tudi metoda za spremljanje njihovega števila in odločanje o nadaljnjem razvoju.

Ključne besede: recepcijska programska oprema, elektronska knjiga gostov, urejevalnik rezervacij, eGost.

Abstract

This thesis describes the motivation for the development, design and operation of the software that implements an electronic guestbook with booking management for small and medium-sized accommodation facilities. The first part covers the presentation of the target user group, description of their work and their needs, analysis of existing solutions on the Slovenian market and a review of the shortcomings that these have and create the need for a new solution. Next, the design of a custom solution that attempts to satisfy this need is presented, followed by the plan for its development and description of the software package eGost, that enables secure guest data internet transmission. The implementation of this custom solution in the form of a free program named Opengost and its functionalities are described in detail. The final part examines the feedback provided by the first users and also describes a method for monitoring their number and making decisions in future development.

Keywords: property management software, electronic guestbook, booking manager, eGost.

Poglavje 1

Uvod

Slovenija je nedvomno slikovita dežela, saj se v njej prepletajo tako Sredozemlje kot Kras, Panonski svet in gorski svet Alp. Razgibanost površja, velika gozdnatost, pestra kulturna dediščina in številne možnosti športnih aktivnosti tvorijo odlično turistično destinacijo za tako domače kot tuje goste.

Zaradi tega ne preseneča, da je turizem v Sloveniji velikega pomena in da predstavlja veliko poslovno priložnost [1]. Skupaj s posrednimi učinki prispeva turizem kar 12,8 odstotkov k celotnemu bruto domačemu proizvodu, predstavlja 8 odstotkov celotnega izvoza in več kot 40 odstotkov izvoza storitev. V turističnih panogah je zaposlenih več kot 110.000 oseb, kar predstavlja več kot 8 odstotkov vseh delovnih mest.

Število turistov, ki se odločijo prenočevati v slovenskih nastanitvenih objektih, z vsakim letom narašča. V primerjavi z letom 2012 se je to število v letu 2013 povečalo za 5 odstotkov, število nočitev, ki so jih ti ustvarili, pa za 3 odstotke. Konstantna rast obiska ustvarja zmeraj večje potrebe po razpoložljivosti v obstoječih prenočitvenih objektih in po nastanku novih.

V tabeli 1.1 so prikazana števila sob in ležišč po vrstah nastanitvenega objekta, ki so bila turistom na voljo v letu 2013. Če želimo ločiti med *majhnimi*, *srednje velikimi* in *velikimi* nastanitvenimi objekti, je potrebno najprej določiti, kaj posamezni pojem predstavlja. Velikost nastanitvenega objekta namreč ni samoumevna količina: prenočitveni objekt ima lahko na primer le

Vrsta nastanitvenega objekta	Št. objektov	Št. sob	Št. ležišč
Hoteli	289	18.801	41.163
Moteli	8	102	224
Penzioni	88	1.013	2.676
Gostišča	160	1.401	3.342
Prenočišča	94	785	1.946
Apartmajska naselja	30	1.719	5.992
Kampi	74	7.119	22.660
Turistične kmetije	335	1.601	4.886
Mladinski hoteli	49	564	2.175
Zasebne sobe, apartmaji, hiše	...	4.776	16.341
Planinski domovi in koč	148	1.162	6.892
Delavski in mladinski domovi	...	1.485	6.725
Drugi nastanitveni objekti	31	163	459
Začasne zmogljivosti in marine	30	2.307	6.696
Skupaj	1.336	42.998	122.177

Tabela 1.1: Število nastanitvenih objektov, sob in ležišč po vrstah nastanitvenega objekta v letu 2013 (povzeto po [2])

10 sob, iz česar bi lahko morda sklepali, da gre za nastanitveni objekt manjše velikosti, a če ima v vsaki sobi 16 ležišč (kot je pogosto pri mladinskih hostlih z dormitoriji), je sklep nedvomno napačen. Podobno je lahko zavajajoče tudi število ležišč: v dveh različnih objektih je lahko na primer 50 sob, pri čemer so v prvem sobe večinoma enoposteljne, v drugem pa velike družinske z najmanj 4 ležišči.

Primerneje je torej velikost objekta določati s kombinacijo *povprečnega števila ležišč na sobo* in *skupnega števila sob*. Če želimo po velikosti ločiti posamezne vrste objektov, pa je potrebno tudi skupno število sob povprečiti s številom objektov iste vrste v Sloveniji. V tabeli 1.2 je predstavljen rezultat izračuna teh vrednosti.

Vrsta nastanitvenega objekta	Povp. ležišč na sobo	Povp. sob na objekt
Hoteli	2,19	65,06
Moteli	2,20	12,75
Penzioni	2,64	11,51
Gostišča	2,39	8,76
Prenočišča	2,48	8,35
Apartmajska naselja	3,49	57,30
Kampi	3,18	96,20
Turistične kmetije	3,05	4,78
Mladinski hoteli	3,86	11,51
Zasebne sobe, apartmaji, hiše	3,42	...
Planinski domovi in koč	5,93	7,85
Delavski in mladinski domovi	4,53	...
Drugi nastanitveni objekti	2,82	5,26
Začasne zmogljivosti in marine	2,90	76,90
Skupaj	2,84	32,18

Tabela 1.2: Povprečno število ležišč na sobo in povprečno število sob na objekt po vrsti nastanitvenega objekta

Vrsta nastanitvenega objekta	Povp. kapaciteta
Hoteli	142,43
Moteli	28,00
Penzioni	30,41
Gostišča	20,89
Prenočišča	20,70
Apartmajska naselja	199,73
Kampi	306,22
Turistične kmetije	14,59
Mladinski hoteli	44,39
Planinski domovi in koč	46,57
Drugi nastanitveni objekti	14,81
Začasne zmogljivosti in marine	223,20
Skupaj	91,45

Tabela 1.3: Povprečna kapaciteta po vrsti nastanitvenega objekta

Definirajmo pojem *povprečna kapaciteta nastanitvenega objekta* kot zmnožek povprečnega števila ležišč na sobo in povprečnega števila sob na vrsto prenočitvenega objekta. Vrednost te količine je dobro merilo za ločitev posameznih tipov objektov po velikosti. V tabeli 1.3 so prikazane vrednosti zmnožka za posamezno vrsto prenočitvenega objekta. Zasebne sobe, apartmaji in hiše ter delavski in mladinski počitniški domovi so izvezeti, saj za izračun ni na voljo vseh potrebnih podatkov.

Pojme *majhen*, *srednje velik* in *velik prenočitveni objekt* definirajmo na naslednji način:

Majhen prenočitveni objekt je objekt s povprečno kapaciteto, ki ne presega vrednosti 30. V tabeli 1.3 temu kriteriju ustrezajo moteli, gostišča, prenočišča, turistične kmetije in druge manj pogoste vrste nastanitvenih objektov. Mednje uvrstimo tudi zasebne sobe, apartmaje in hiše, ki jih v tabeli ni;

Srednje velik prenočitveni objekt je objekt s povprečno kapaciteto med 30 in 100. V tabeli 1.3 ta kriterij izpolnjujejo penzioni, mladinski hoteli, planinski domovi in kočje;

Velik prenočitveni objekt ima povprečno kapaciteto večjo od 100. V tabeli 1.3 k velikim objektom sodijo hoteli, apartmajska naselja, kampi ter začasne zmogljivosti in marine. Mednje uvrstimo tudi delavske in mladinske domove, ki jih v tabeli ni.

S pomočjo stolpca *Št. objektov* v tabeli 1.1 in pravkar definirane klasifikacije velikosti tipov nastanitvenih objektov lahko izračunamo, kolikšen delež vseh objektov v Sloveniji predstavlja posamezni velikostni razred. Izračun pokaže, da majhni nastanitveni objekti predstavljajo 47 odstotkov, srednji 21 odstotkov in veliki 32 odstotkov vseh nastanitvenih objektov v Sloveniji. Skoraj polovica nastanitev v Sloveniji je torej majhnih, približno petina jih je srednje velikosti, slabo tretjino vseh pa predstavljajo večji objekti.

Majhni in srednje veliki nastanitveni objekti skupaj predstavljajo dobri dve tretjini vseh nastanitev. V nadaljevanju besedila združimo oba velikostna razreda v enega in ga poimenujmo *manjši nastanitveni objekti*. Z vidika podjetij, ki razvijajo računalniško programsko opremo, predstavlja ta velikostni razred veliko in zanimivo ciljno skupino. Pričakovali bi, da je zaradi tega na slovenskem trgu veliko računalniških produktov, s katerimi bi si lahko zaposleni v manjših nastanitvenih objektih olajšali delo, in da je konkurenčnost med produkti velika, a se izkaže, da temu ni tako.

V Sloveniji je ponudba programske opreme za manjše objekte zelo šibka. Nasprotno je programskih rešitev za velike nastanitvene objekte veliko, še posebno pestra pa je izbira profesionalnih tujih rešitev, prilagojenih slovenskim zahtevam.

V diplomskem delu so bralcu najprej predstavljeni potek recepcijskega dela in potrebe v manjših nastanitvenih objektih. Ob pregledu obstoječih rešitev avtor ugotavlja, da obstoječe rešitve bodisi ne zadovoljujejo potreb ali pa da so predrage in prezapletene za ciljnega uporabnika.

Poglavitni cilj diplomskega dela je razvoj nove, sodobne in brezplačne rešitve, namenjene manjšim nastanitvenim objektom, in posledično zapolniti tržno nišo, ki na tem področju obstaja. Avtor v delu opiše lastno rešitev od njene zasnove do implementacije in predstavi odzive prvih uporabnikov.

Poglavje 2

Potrebe manjših nastanitev

Recepcijsko delo je lahko zelo širok pojem, ki obsega najmanj [3]:

- sprejem povpraševanj in rezervacij preko telefona, e-pošte, klasične pošte, telefaksa ali v živo,
- izvrševanje procedur ob prihodu in odhodu gostov,
- izbiranje sob in izročanje ključev gostom,
- načrtovanje zasedenosti in urejanje obstoječih rezervacij tako, da se razpoložljivost v prihodnosti maksimizira,
- pripravo poročil, ki jih je nastanitev zakonsko dolžna izdelovati in pošiljati ustreznim organom,
- prijavljanje osebnih podatkov gostov na policijo,
- pripravo poročil za lastne potrebe nastanitve (na primer seznam sob za čiščenje, seznam dnevnih prihodov in podobno),
- pripravo računov in sprejem plačil,
- nudenje raznovrstnih informacij o storitvah, kraju in aktualnih dogodkih ter

- razreševanje pritožb gostov in težav, v katerih se ti znajdejo.

Pri nastanitvah, ki obsegajo le nekaj sob, je recepcijsko delo lahko na prvi pogled zelo različno od tistega v večjih nastanitvah, vendar kljub temu zajema enake naloge in procese, le v manjšem obsegu. Pri le-teh vse naloge običajno sočasno opravlja le ena oseba, kateri so pogosto dodeljene tudi dodatne naloge, ki niso del recepcijskega dela (na primer strežba pijače, računovodstvo in podobno). Pri večjih nastanitvah je lahko za opravljanje naštetih nalog zadolženih več oseb, pri čemer vsaka izmed njih opravlja le natanko določeno podmnožico nalog.

Večina izmed naštetih nalog je *administrativne narave* in jih je mogoče olajšati z namensko programsko opremo, medtem ko jih je nekaj *neadministrativne narave* (kot na primer reševanje pritožb), kjer to običajno ni mogoče.

Namensko programsko opremo za olajšanje, optimizacijo, povečevanje učinkovitosti ali nadzor recepcijskega dela imenujemo *recepcijska programska oprema* (pri večjih nastanitvah tudi *hotelski informacijski sistem* ali pogosteje v angleščini *Property management system* oz. *PMS* [3]). Ta je lahko implementirana v obliki *modulov* (če digitalizira in avtomatizira le podmnožico nalog administrativne narave) ali v obliki *celovite rešitve* (če poskuša digitalizirati in avtomatizirati vse naloge administrativne narave). Pri zelo velikih hotelih je lahko recepcijska programska oprema le del še večjega informacijskega sistema celotne hotelske verige.

Moduli so lahko medsebojno povezljivi ali samostojni. V praksi jih glede na naloge, katerim so namenjeni, delimo na naslednje:

Elektronska knjiga gostov (*electronic guestbook*) je osnovni modul, ki nadomešča fizično knjigo gostov, v katero mora nastanitev vnašati osebne podatke gostov, ki prenočujejo. Modul na osnovi vpisanih podatkov pripravlja prijavnice, ki jih nato preko spleta pošilja informacijskemu sistemu policije. Poleg tega lahko knjiga gostov služi kot arhiv podatkov o preteklih prenočevanjih in ponuja orodja za izdelavo statističnih poročil in poročil o pobrani turistični taksi;

Urejevalnik rezervacij (*Central Reservation System* oz. *CRS*) skrbi za hrambo povpraševanj in rezervacij ter ponuja vizualna orodja za njihov vnos, urejanje in pregled ter ugotavljanje zasedenosti nastanitve. Vsebuje bogato bazo preteklih in novih rezervacij, na osnovi katerih lahko izdeluje izčrpna poročila;

Upravljalac kanalov (*Channel Manager*) je modul, ki skrbi za sinhronizacijo razpoložljivosti na različnih kanalih in prenos rezervacij z zunanjih kanalov v lastni urejevalnik rezervacij. *Kanal* predstavlja vir rezervacij, ki je pogosto v obliki spletnih aplikacij. Primeri najbolj razširjenih zunanjih kanalov v Sloveniji so *Booking.com*, *Hostelworld* in *Airbnb*;

Blagajna (*Point-of-Sale*) omogoča izdajo računov, pripravo predračunov in ponudb. Skladna mora biti z zakonodajo, ki se pogosto spreminja, in ponujati mora izvoz podatkov v primerni obliki za nadaljnjo obdelavo v računovodstvu;

Dodatna orodja so običajno manj obsežni moduli, ki ne sodijo v nobeno izmed zgoraj naštetih skupin.

Zgornji opisi so zgolj površinski in presplošni, da bi razvijalcu recepcijske programske opreme služili kot vodilo. Pri razvoju je potrebno dobro poznati realno recepcijsko delo, številne težko predvidljive situacije, potrebe in zahteve del, ki niso del recepcijskega, ter zahteve, ki so predpisane z zakoni. Zaradi tega je za razvijalca zelo koristno, če spremlja ali se tudi sam udeleži klasičnega recepcijskega dela, v katerem se namenska računalniška oprema ne uporablja ali se uporablja le v zelo majhnem obsegu.

2.1 Administrativno recepcijsko delo brez namenske računalniške opreme

Podpoglavje podrobno opisuje potek administrativnega recepcijskega dela v manjših nastanitvenih objektih, kjer se namenska recepcijska programska

oprema ne uporablja. Poudariti je treba, da to ne pomeni, da se v takih objektih računalniških programov ne uporablja - uporabljajo se namreč lahko splošna orodja, ki niso bila specifično razvita za recepcijsko delo, vendar ga kljub temu poenostavijo. Če se v objektu podatke gostov vpisuje na primer v tabelo s programom Excel in nato podatke prepíše v fizično knjigo gostov, ne moremo reči, da je Excel recepcijska programska oprema.

2.1.1 Zapis rezervacij

Vsak nastanitveni objekt, ne glede na velikost, velik del nočitev pridobiva v obliki najavljenih nočitev v prihodnosti, čemur običajno pravimo rezervacija. Pri namestitvi gosta, ki pred prihodom svojega bivanja ne najavi, lahko prav tako govorimo o rezervaciji - za razliko od rezervacije v prihodnosti je ta opravljena v živo v trenutku dogovora o najemu enote.

Rezervacija je torej pojem, ki zajema tako najavljene nočitve v prihodnosti kot nočitve, za katere se gost brez najave dogovori ob prihodu, in nočitve, ki so bile že opravljene v preteklosti.

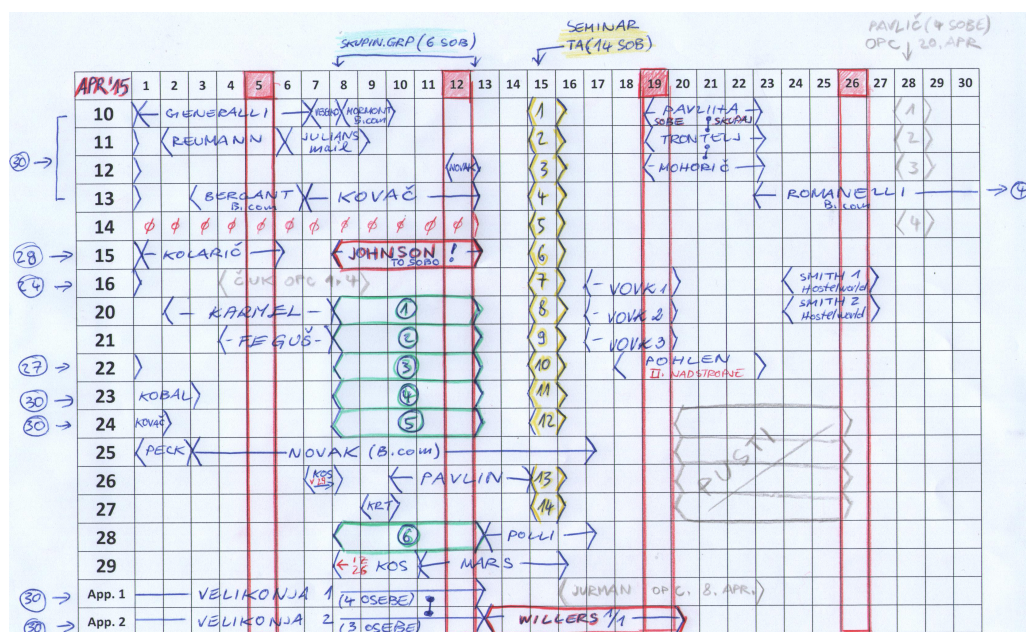
Zapis podatkov o rezervacijah je v vsaki obliki nastanitve primarnega pomena. Ta je lahko v obliki seznama, tabele, mreže ali v obliki kakšne druge vizualne strukture. Pri sobodajalcih, ki razpolagajo z eno samo sobo, je lahko preprost spisek rezervacij popolnoma zadosten. Pri nastanitvah z več sobami pa taka oblika zapisa ni primerna, saj je s seznamom težko razbrati proste termine posameznih sob, rezervacije spreminjati ali iskati.

V praksi veliko manjših nastanitvenih objektov rezervacije hrani v obliki *mreže* (včasih poimenovana tudi *gantogram*, *koledar* ali *časovnica*). Mrežo sestavlja prazna tabela, v kateri posamezne vrstice predstavljajo enote nastanitvenega objekta, ki jih je možno rezervirati (sobe, deli sob ali posamezna ležišča), stolpci pa dneve v mesecu. Tabela se nato za vsak mesec natisne na papir primerno velikega formata in se vanjo vpisuje rezervacije. Primer take mreže je predstavljen na sliki 2.1.

Bralec lahko hitro ugotovi, da je taka vrsta zapisa veliko bolj pregledna od seznama ali drugih manj vizualnih oblik. Obenem pa lahko tudi opazi,

2.1. ADMINISTRATIVNO RECEPCIJSKO DELO BREZ NAMENSKE RAČUNALNIŠKE OPREME

11



Slika 2.1: Primer fizične mreže za zapis rezervacij v hotelu z 19 enotami v mesecu aprilu 2015

	1	2	3	4	5	6	7	8	9	10	11	12	13
10	X	GENERALI					X	X	X	X			
11		REUMANN					X	JULIANS					
12												X	NOVAK
13				BERCLANT			X	KOVAC					
14	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ	φ
15	X	KOLARIC						X	X	X	X	X	X

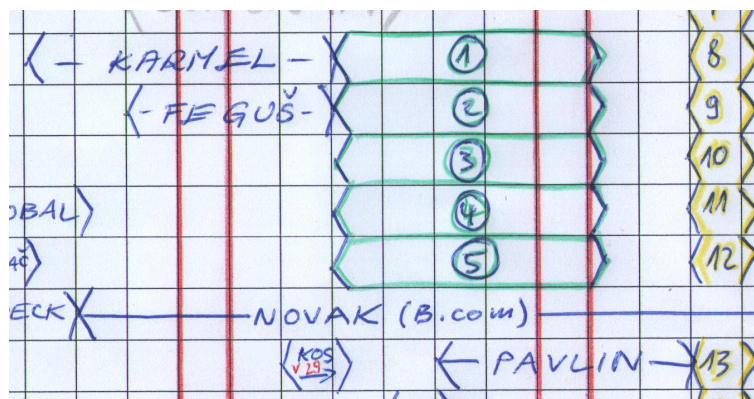
Slika 2.2: Povečava dela mreže s slike 2.1

da je preglednost v terminih večje zasedenosti precej slaba.

Na sliki 2.2 je prikazana povečava dela mreže s slike 2.1. V mreži posamezno rezervacijo enote predstavljajo celice med znakoma < in >. Med znakoma je zapisano ime rezervacije, pogosto pa tudi dodatni podatki (na primer kanal, preko katerega je bila rezervacija prejeta, izpostavljene zahteve in podobno). Znak < se nahaja v celici, ki označuje datum prihoda gostov, znak > pa v celici, v katerem gostje odhajajo.

Uporabljene barve v mreži imajo natanko določen pomen. Z rdečo barvo se v mreži običajno opremi tiste dele, kjer mora biti oseba, ki z mrežo dela, še posebej pozorna. V primeru na sliki 2.2 so z rdečo barvo označene nedelje, saj je veliko nastanitvenih objektov bolj zasedenih ob koncu tedna in je potrebno pri teh bolj paziti na zasedenost. Rezervacija sobe 15 na ime Johnson med 8. in 13. aprilom je obrobljena z rdečo barvo, saj je gost ob kreaciji rezervacije zahteval natanko to sobo, čemur pravimo tudi *fiksna rezervacija*. Soba 14 je do 13. aprila označena z rdečimi znaki ϕ , saj enote iz nekega razloga ni mogoče rezervirati (v sobi na primer ne deluje gretje).

Z ostalimi barvami so označene *skupinske rezervacije*. Enaka barva v enakem terminu označuje vse rezervacije enot, ki taki rezervaciji pripadajo. Ker imena gostov, ki bodo bivali v posamezni enoti, pri skupinskih rezervacijah

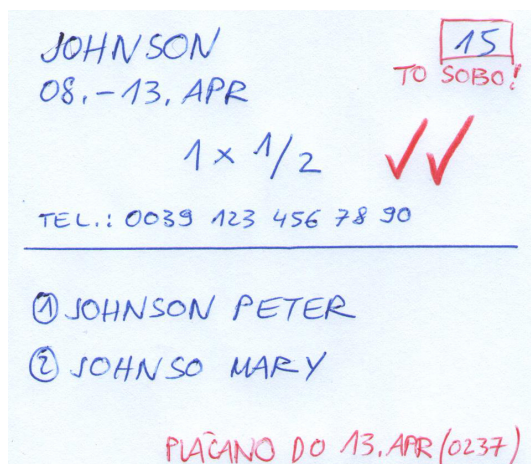


Slika 2.3: Povečava dela mreže s slike 2.1

velikokrat niso znana, so te rezervacije oštevilčene, nad mrežo pa sta zapisana ime skupine in število rezerviranih enot. Na sliki 2.3 lahko bralec prepozna 2 skupinski rezervaciji, pri čemer je prva označena z zeleno barvo in obsega 5 enot, druga pa je označena z rumeno barvo in obsega večje število enot.

S sivo barvo (s svinčnikom) se v mrežo vpisuje rezervacije (povpraševanja), ki še niso potrjene in jim pravimo tudi *opcijske rezervacije*. Pri takih rezervacijah je potrebno pod imenom zapisati tudi datum, do katerega mora gost rezervacijo potrditi. V primeru pretečenega roka potrditve se rezervacijo lahko iz mreže izbriše.

Velikost celic v mreži je pogosto premajhna, da bi bilo v mrežo mogoče vpisati vse podatke o rezervacijah. Zaradi tega se poleg vpisa v mrežo rezervacije in njihove podrobnosti zapisuje tudi na drugačen način. V ta namen se lahko na primer uporabljajo beležke, zvezki, planerji, digitalni tekstovni dokumenti ali druge oblike zapisa besedila. Nekatere nastanitvene enote podrobnosti posamezne rezervacije enot zapisujejo na liste manjšega papirnatega bloka in nato liste iz bloka iztrgajo ter vstavijo v zbirno mapo, v kateri so rezervacije grupirane po datumih prihoda. Slika 2.4 prikazuje primer takega načina zapisa.



Slika 2.4: Primer zapisa dodatnih podatkov rezervacije, za katere v mreži ni dovolj prostora

2.1.2 Urejanje rezervacij

Z uporabo mreže ali podobnih načinov zapisa obstaja v vsakem trenutku za vsako rezervacijo predvidena enota, v katero bodo gostje ob prihodu nameščeni. V praksi le majhen delež gostov pristane v enoti, ki jim je bila dodeljena ob kreaciji rezervacije. Razlogov je več, najpogostejši pa so naslednji:

- rezervacijam v prihodnosti oseba, ki je za to zadolžena, dodeli druge enote z namenom maksimizacije razpoložljivosti (pri uporabi mreže lahko na to gledamo kot na defragmentacijo),
- gost v obdobju med kreacijo rezervacije in prihodom spremeni zahteve, določi nov datum prihoda, skrajša ali podaljša bivanje ter spremeni število enot, ki jih potrebuje,
- enota, za katero že obstaja dodeljena rezervacija, ni več na voljo ali jo fiksno zahteva drugi gost.

Pri vsaki izmed zgornjih sprememb je potrebno rezervacijo v mreži premakniti. *Premik* pomeni izbris rezervacije iz mreže in ponovni vpis v druge

celice. Pri fizični mreži zahteva ta proces uporabo pripomočkov za brisanje (na primer belila, radirke in podobnih). Pri urejanju večjega števila rezervacij je potrebno biti zelo pazljiv in si njihove podatke pred izbrisom iz mreže prepisati na drug medij. Proces lahko hitro postane zahteven in naporen, še posebej pri maksimiziranju razpoložljivosti, saj je do dobrega rezultata mogoče priti le s poskušanjem in ponovnim popraviljanjem.

Urejanje rezervacij je torej zahteven del recepcijskega dela, pri katerem sta digitalizacija in avtomatizacija z namensko programsko opremo zelo dobrodošli in skorajda nujni.

2.1.3 Prijava gostov

V Republiki Sloveniji je potrebno vsakega gosta ob prihodu zaprositi za osebni dokument in v knjigo gostov prepisati naslednje podatke:

- priimek,
- ime,
- datum rojstva,
- kraj rojstva,
- državljanstvo,
- datum prihoda in
- datum odhoda.

Te podatke je nastanitev dolžna posredovati policiji, pri čemer lahko to stori na naslednje načine:

- z izpolnitvijo namenskega obrazca, s katerim je mogoče prijaviti 1 odraslo osebo in več mladoletnih oseb v spremstvu, ter oddajo obrazca policiji (primer na sliki 2.5),

- z vpisom podatkov vseh gostov v fizično knjigo gostov (trdo vezana knjiga formata A4), kopiranjem strani iz knjige in oddajo kopij na policijo ali
- z uporabo programskega paketa eGost, ki omogoča varno oddajo prijavnic preko spleta, ob pogoju, da nastanitev uporablja elektronsko knjigo gostov, ki je s paketom združljiva.

Nastanitveni objekt je zakonsko obvezan vsakega gosta, ki prenočuje, prijaviti najkasneje 12 ur po prihodu, kar pomeni, da je včasih potrebno prijavnice oddajati dvakrat dnevno. Pri recepcijskem delu brez namenske programske opreme je torej prijava gostov lahko zelo dolgotrajen in neprijeten proces.

Ob spremembi bivanja gostov (skupina gostov se na primer odloči skrajšati ali podaljšati bivanje) je potrebno goste ponovno prijaviti, kar povzroči še več dodatnega dela.

Elektronska knjiga gostov, ki digitalizira in avtomatizira ta proces, je torej, enako kot urejevalnik rezervacij, eden izmed najnujnejših modulov recepcijske programske opreme.

Poleg naštetih podatkov, ki jih je nastanitveni objekt dolžen prijavljati policiji, pa je potrebno hraniti tudi naslednje:

- spol,
- naslov stalnega bivališča,
- naziv javne listine, s katere so bili pridobljeni podatki, ter
- številko javne listine.

Način hrambe teh podatkov ni specificiran in je lahko poljuben. Nekatere elektronske knjige gostov vnos teh podatkov omogočajo, nekatere pa ne. V slednjem primeru in v primeru, da nastanitveni objekt ne uporablja elektronske knjige gostov, se mora objekt sam odločiti, kako bo te podatke shranjeval.

2.1. ADMINISTRATIVNO RECEPCIJSKO DELO BREZ NAMENSKE RAČUNALNIŠKE OPREME

17

Priloga št. 10

ZIG STANOVALCA
LANDLORD'S SEAL
CACHET DU LOGEUR
TMBO DI LOCATORE
STEMPEL DES VERMIETERS
A SZALLÁSADO PECSÉTJE

ZAPOREDNA ŠTEVILKA PRIJAVNICE
REFERENCE NUMBER OF REGISTRATION FORM
NO. DE SUITE DE LA DECLARATION
NO. D'ORDINE DI DICHIARAZIONE
SERIENNUMMER DES ANMELDESCHHEINES
BEJELENTŐLAP SORSZÁMA

000734

PRIJAVA- ODJAVA GOSTA*
REGISTRATION - REVOCATION OF GUEST*
DÉCLARATION ARRIVÉE - DÉPART DE L'HÔTE*
REGISTRAZIONE - PARTENZA DELL'OSPITE*
GÄSTEANMELDUNG UND - AMBELDUNG*
VENDÉG BEJELENTÉSE - KIJELENTÉSE*

*Ustrezno obkrožite!
*Circle the appropriate answer!
*Entourez la variante concernée!
*Accerchiare la risposta relativa!
*Entsprechendes bitte ankreuzen!
*Kiválasztva be a megjelölt!

NOVAK JANEZ
PRIIMEK IN IMEN/NAME AND FAMILY NAME/NOM ET PRÉNOM/NOME E COGNOME/ FAMILIENNAME, VORNAME/CSALÁDI ÉS UTÓNÉV

0101900500100
EMŠO (oznaka datum rojstva in spol, če ta ni določena)/EMŠO (or date of birth and sex if no EMŠO has been assigned)/ No. d'immatriculation (ou, à défaut de celui-ci déterminé, date de naissance et sexe)/CAUCE (o la date di nascita e sesso, se CAUC non era assegnato)/PERSONENKENNZIFFER (bzw. Geburtsdatum und Geschlecht, falls die Personenkennziffer nicht vorhanden)/SZEMÉLYI SZÁM (illetve születési dátum és nem, ha ez nincs megadva)

LJUBLJANA
KRAJ ROJSTVA/PLACE OF BIRTH/LIEU DE NAISSANCE/LUOGO DI NASCITA/GEBURTSORT/SZÜLETÉSI HELY

SLOVENSKA ULICA 1, 1000 LJUBLJANA
STALNO PREBIVALISČE (občina, naselje, ulica, hišna številka z dodatkom, za tuje tudi država)/PERMANENT RESIDENCE (municipality, town, street, house number and any appendices; for those registered abroad, also country)/DOMICILE(commune, localité, rue, numéro de la maison avec supplément éventuel, pour les étrangers également l'indication du pays)/RESIDENZA PERMANENTE (comune, luogo, via, numero di casa con eventuali supplementi, i stranieri devono indicare anche oro paese)/HAUPTWOHNSTZ (Gemeinde, Ortschaft, Straße, Haus-No.mit ev. Zusatz; bei Ausländern auch das Land angeben)/ÁLLANDO LAKOHELY (község, település, utca, házszám, külföldi esetében az ország is)

SLOVENSKO
DRŽAVLJANSTVO/CITIZENSHIP/NATIONALITÉ/CITTADINANZA/STAATSBURGERSCHAFT/ALLMÖPÖLÖRSÄG

123456789
ŠTEVILKA JAVNE LISTINE IN ORGANA, KI JO JE IZDAL/NUMBER OF PUBLIC DOCUMENT AND THE BODY WHICH ISSUED IT/NO. D'ACTE ET AUTORITÉ QUI L'A DELIVRÉ/NUMERO DI DOCUMENTO E L'AUTORITÀ DI RILASCIO/NUMMER DER ÖFFENTLICHEN URKUNDEN; AUSSTELLUNGSBEHÖRDE/KÖZÖKIRAT SZÁMA ES A KIÁLLÍTÓ HATÓSÁG

PRIJAVLJAM TUDI NASLEDNJE MLADOLETNE OSEBE (posameznike, mlajše od 15 let)
REGISTRATION OF MINORS (persons younger than 15)
JE DÉCLARE ÉGALEMENT LES PERSONNES MINEURES SUIVANTES (individus qui n'ont pas atteint l'âge de 15 ans)
DICHIARO ANCHE LE SEGUENTI PERSONE MINORI (persone sotto 15 anni)
ANMELDUNG DER MINDERJÄHRINGEN JUGENDLICHEN (Personen unter 15 Jahren)
BEJELENTÉSEM AZ ALÁBBI KISKORÚ SZEMÉLYEKET IS (15 évnél fiatalabb egyének)

PRIIMEK IN IME SURNAME AND GIVEN NAMES NOM DE FAMILLE ET PRÉNOM NOME E COGNOME FAMILIENNAME, VORNAME CSALÁDI ÉS UTÓNÉV	EMŠO oznaka datum rojstva EMŠO or date of birth No. d'immatriculation ou date de naissance CAUC o la data di nascita Personenkennziffer/Geburtsdatum SZEMÉLYI SZÁM illetve születési dátum	KRAJ ROJSTVA PLACE OF BIRTH LIEU DE NAISSANCE LUOGO DI NASCITA GEBURTSORT SZÜLETÉSI HELY	DRŽAVLJANSTVO CITIZENSHIP NATIONALITÉ CITTADINANZA STAATSBURGERSCHAFT ALLMÖPÖLÖRSÄG
<u>NOVAK MIHA</u>	<u>01.01.2011</u>	<u>LJUBLJANA</u>	<u>SLOVENSKO</u>
<u>NOVAK NINA</u>	<u>01.01.2007</u>	<u>LJUBLJANA</u>	<u>SLOVENSKO</u>

08.04.2015 09.04.2015

DATUM PRIJAVE
DATE OF REGISTRATION
DATE DE DECLARATION ARRIVÉE
DATE DI REGISTRAZIONE
ANREISEDATUM
BEJELENTÉSI DÁTUM

DATUM ODJAVE
DATE OF REVOCATION
DATE DE DÉPART
DATE DI PARTENZA
ABREISEDATUM
KIJELÉNTÉSI DÁTUM

PODPIS PRIJAVITELJA
SIGNATURE BY PERSON REGISTERING
SIGNATURE DU RÉGISTRÉ
FIRMA DI ISCRIVENTE
UNTERSCHRIFT DES ANMELDERS
BEJELENTŐ ALÁÍRÁSA

Slika 2.5: Primer izpolnjenega obrazca, s katerim je na policijo mogoče prijaviti 1 odraslo osebo in več mladoletnih oseb v spremstvu

2.1.4 Vodenje statistike in priprava poročil

Nastanitveni objekti so v Sloveniji dolžni voditi statistiko prihodov, nočitev in oddanih enot ter podatke na mesečni in letni ravni sporočati ustreznim organom:

- na Statistični urad je potrebno vsak mesec poslati izpolnjen **Vprašalnik za mesečno statistično raziskovanje Prihodi in prenočitve turistov** (obrazec TU/M), v katerem je potrebno navesti število prihodov in število nočitev po narodnosti za pretekli mesec ter skupno število prodanih enot. Vzorčna stran obrazca je vidna na sliki 2.6,
- na Statistični urad je potrebno vsako leto poslati izpolnjen **Vprašalnik za statistično raziskovanje Prihodi in prenočitve turistov** (obrazec TU/M-DOD), v katerem je potrebno navesti število prihodov in število nočitev po narodnosti v celotnem preteklem letu ter skupno število prodanih enot,
- na občino, v kateri se nastanitev nahaja, je potrebno vsak mesec oddati dokument, v katerem so navedeni število pobranih turističnih taks (po vrsti takse), skupni znesek ter statistika prihodov in nočitev, ločena na domače in tuje goste.

Poleg tega so nastanitve v nekaterih občinah dolžne na dnevni ravni lokalnemu turističnemu centru sporočati število domačih in tujih gostov, ki so v pretekli noči prenočevali.

Da je izdelava naštetih statističnih poročil sploh mogoča, mora nastanitev pripraviti najmanj tabelo, v katero lahko vpisuje dnevno število prihodov in prenočitev po narodnosti, tabelo, v katero vpisuje število pobranih turističnih taks po vrsti takse ob vsakem prejetem plačilu, ter tabelo, v katero vpisuje število prodanih enot na dan. Tabele je nato potrebno izpolnjevati redno in dosledno ter jih pri vsaki spremembi prenočitve (gost na primer skrajša ali podaljša bivanje) popravljati. Iz tabel je nato mogoč izračun vseh potrebnih podatkov za tvorbo predpisanih poročil.

Sklop B **Prihodi in prenočitve turistov**

B1 Vpišite število turistov in število njihovih prenočitev v opazovanem mesecu, in sicer glede na državo, v kateri ima posamezni turist prebivališče:

- Upoštevajte samo prihode turistov v opazovanem mesecu. Če je npr. turist prišel v nastanitveni objekt ob koncu januarja in ostal v njem še v začetku februarja, se njegov prihod upošteva samo enkrat, in to samo v podatkih za mesec prihoda, tj. za januar.
- Podatki o številu prenočitev se vnesejo glede na dejansko število prenočitev v opazovanem mesecu. Če je turist v nastanitveni objekt prišel npr. 29. januarja in v njem prenočil 7-krat, potem v vprašalniku za januar upoštevate 3 prenočitve, v vprašalniku za februar pa 4 prenočitve.
- Prenočitev se vedno nanaša na prejšnji dan. Primer: noč med 31. januarjem in 1. februarjem se šteje v januar.
- Med turiste šteje tudi otroke, tudi takrat, kadar ti nimajo posebnega ležišča.
- V podatkih ne upoštevajte oseb, ki so v Sloveniji na začasnem delu in v Sloveniji prejmejo plačilo za opravljeno delo.
- Poleg podatkov o počitniških turistih morate vpisati tudi podatke o poslovnih turistih, ki so obiskali Slovenijo z namenom udeležbe na sestanku, konferenci, izobraževanju, udejstvovanju na zabavni, kulturni ali športni prireditvi ipd.

Država, v kateri ima turist prebivališče	Število turistov	Število prenočitev (skupaj)
EVROPA		
Slovenija 3 705		
Avstrija 040		
Belgija 006		
Bolgarija 100		
Bosna in Hercegovina 070		
Ciper 196		
Češka republika 203		
Črna gora 499		
Danska 208		
Estonija 233		
Finska 246		
Francija 250		
Grčija 300		
Hrvaška 191		
Irski 372		
Islandija 362		

Slika 2.6: Vzorčna stran z obrazca TU/M

Iz zgornjega opisa je očitno, da je vodenje statistike brez namenske programske opreme v nastanitvah z več kot eno enoto lahko izredno zahtevno in zamudno delo. Nekatere nastanitve proces delno olajšujejo z uporabo računalniških programov za obdelavo tabelaričnih podatkov (na primer Excel), s pomočjo katerih je mogoče avtomatizirati končne izračune, a je vnos vsakega prihoda, vsake nočitve, vsake pobrane takse in vsake prodane enote še zmeraj potrebno opravljati ročno.

Bralec se bo spomnil, da je državljanstvo gosta eden izmed obveznih podatkov, ki se vnašajo v knjigo gostov. Elektronske knjige gostov imajo zaradi tega na voljo vse potrebno, da lahko iz vpisov gostov same generirajo statistična poročila prihodov in nočitev po narodnosti. Če elektronska knjiga gostov omogoča vpis enote, v katero je gost nameščen, in vpis vrste turistične takse, ki jo je gost dolžen plačati, potem lahko dobra knjiga gostov izdela tudi poročilo o pobrani turistični taksi in v statistično poročilo vključi število prodanih enot.

Poleg poročil, ki so zakonsko predpisana, pa je v praksi pogosto potrebno za lastne potrebe izdelati tudi druge vrste poročil:

- poročilo o dnevnem stanju enot, iz katerega je razvidno, iz katerih enot gostje odhajajo, v katerih ostajajo in v katerih ni nameščenih gostov (poročilo je potrebno pri čiščenju sob in pri ugotavljanju morebitnih nepravilnosti),
- poročilo o dnevnih prihodih, ki ob prihodu gostov omogoča lažjo in hitrejšo namestitve v predvidene enote,
- poročilo o prostih enotah in o številu dni do naslednje rezervacije, ki obstajajo zanje (tako poročilo omogoča hitro in optimalno nameščanje nenajavljenih gostov), ter
- druga poročila, ki so specifična za posamezne vrste nastanitvenih objektov.

SOBA	TERMIN	OPOMBE	SOBA	TERMIN	OPOMBE
10	7./8.		20	2./8.	
11	6./9.		21	4./8.	
12			22		
13	7./13.		23		
14	X	OKVARA GREJA	24		
15			25		
16			26	7./11.	SEUTEV V 29
			27		
App. 1	30./13.		28		
App. 2	30./13.		29	7./11.	← 12 26

Slika 2.7: Primer ročno izdelanega poročila o dnevnem stanju sob

Podobno kot pri zakonsko predpisanih poročilih velja, da imata elektronska knjiga gostov in urejevalnik rezervacij dovolj podatkov, da lahko taka poročila generirata samodejno. Nastanitveni objekt, ki recepcijske programske opreme ne uporablja, pa mora zopet ta proces opravljati ročno. Na sliki 2.7 je prikazan primer ročno izdelanega poročila o dnevnem stanju sob za 8. april na podlagi mreže s slike 2.1.

2.1.5 Upravljanje kanalov

Zunanji kanali predstavljajo enega najpomembnejših virov rezervacij in v marsikaterem nastanitvenem objektu prispevajo več novih rezervacij kot e-pošta, pošta, telefon, faks in dogovori v živo skupaj. Ti so pogosto v obliki spletnih strani, preko katerih lahko potencialni gostje iščejo nastanitvene objekte, si ogledujejo njihove lastnosti in razpoložljivost ter opravijo spletno rezervacijo. Objekt, ki se odloči za uporabo zunanjega kanala, se mora najprej registrirati, pripraviti javni profil in s kanalom skleniti alotmajsko pogodbo, s katero se določi vrsta in minimalna količina enot, s katerimi kanal lahko razpolaga. Kanali običajno za vsako pridobljeno rezervacijo od objekta zahtevajo dogovorjen delež od plačila ali pa svoje storitve zaračunavajo v obliki fiksnih mesečnih naročnin.

Za čimvečji izkoristek kanalov in v izogib nastanku večjega števila rezer-

vacij, kot jih nastanitveni objekt lahko sprejme (*overbooking*), mora objekt z uporabo administrativnih orodij dosledno urejati razpoložljivost. Kanali, ki so v obliki spletnih strani, običajno objektom nudijo spletna grafična orodja, s katerimi je možno po tipih enot za vsak datum določiti število razpoložljivih enot, spreminjati cene nočitev, nastavljati najmanjše dovoljeno število dni rezervacij in veliko drugega.

Kanali v splošnem ponujajo programske vmesnike, s katerimi jih je mogoče povezati s centralnim rezervacijskim sistemom (*central reservation system* ali *CMS*), ki predstavlja jedro celovite recepcijske programske opreme. V tem primeru objektu ni potrebno razpoložljivosti urejati ročno, saj centralni rezervacijski sistem skrbi za to sam. Taka rešitev je torej idealna, vendar je s programerskega vidika kompleksna in posledično za manjše objekte zelo draga.

Veliko manjših objektov zato razpoložljivost raje ureja ročno. Pri manjšem številu enot to ni tako zahteven proces, kot se morda na prvi pogled zdi, saj razpoložljivosti na kanalu ni potrebno popolnoma sinhronizirati z realno. Pomembno je le, da na kanalu ni ponujenih več enot, kot jih je v resnici na voljo.

Urejevalnik rezervacij, ki ni povezljiv z zunanjimi kanali, lahko kljub temu delo olajša s pripravo poročil o zasedenosti posameznih tipov enot v izbranem časovnem okviru. Uporabniku tako ni potrebno za vsak termin prostih enot prešteti, pač pa v poročilu poiskati bolj zasedene termine in le zanje uskladiti razpoložljivost na kanalu.

2.1.6 Priprava računov

K že opisanim administrativnim recepcijskim delom, ki so lahko obsežna in časovno zahtevna, je potrebno dodati tudi pripravo računov. Račun se lahko pripravi pred opravljeno storitvijo (v obliki predplačila), ob prihodu, med bivanjem gostov v objektu (pogosto zahtevano pri dolgih rezervacijah), ob odhodu ali kasneje (običajno pri rezervacijah, za katere je bila izdana naročilnica). Za pripravo računa so potrebni naslednji podatki:


- število nočitev, ki so bile ali bodo opravljene,
- cenik nočitev po tipih enot in številu oseb v njih ali dogovorjena cena nočitev,
- število in vrsta turističnih taks za vse osebe, ki so ali bodo prenočevale,
- morebitni dogovorjeni popusti in
- morebitni dodatni stroški.

Oseba, ki pripravlja račun, mora vse naštetu izraziti v obliki posameznih postavk (storitev ali artiklov), izračunati skupni znesek, osnovo in vrednost znižane stopnje DDV (velja za nočitve), osnovo in vrednost splošne stopnje DDV (velja za ostale storitve razen turistične takse, ki ni obdavčena) ter račun opremiti s:

- podatki prejemnika plačila (podatki nastanitvenega objekta),
- podatki plačnika,
- krajem in datumom izdaje računa,
- zaporedno številko računa,
- datumom opravljanja storitev (ki ga je potrebno zapisati kot termin med datumom prihoda in datumom odhoda) in
- lastnim podpisom (ter morebitnim žigom podjetja).

Brez namenske recepcijske programske opreme proces zahteva dobro predpripravo. Ker priprava računa zahteva najmanj nekaj minut nemotenega dela, je v tem primeru dobra praksa račun pripraviti takoj, ko je to mogoče. Primer ročno izdelanega računa z uporabo običajnega samokopirnega računskega bloka je prikazan na sliki 2.8.

Uporaba blagajniške programske opreme lahko opisan proces avtomatizira na dveh nivojih:

HOTEL FRI LJUBLJANA 1000 LJUBLJANA Davčna št. SI 123456789					Račun		
Prejemnik					kraj LJUBLJANA št. 318		
Datum odpošiljanja blaga oz. opravljanja storitev*					datum 01.01.2015		
Datum odpošiljanja blaga oz. opravljanja storitev*					31.12.2014 - 01.01.2015		
Zap. št.	Vrsta blaga oziroma storitve	Količina	Cena na enoto* (brez DDV)	Vrednost* (brez DDV)	Stopnja	DDV* Znesek	Vrednost z DDV
1	2	3	4	5 (Brd)	6	7 (Brd)	8 (5+7)
1.	NOČITEV	1	82,19	82,19	9,5	7,81	90,00 €
2.	TURISTIČNA TAKSA	2	1,01	2,02	0	0	2,02 €
							
Zbrana št. DDV					7,81 €		od osnove
Splošna št. DDV					82,19 €		od osnove
Prodajna vrednost					92,02 €		
Obrn. 5,30							

* Na splošno določeno v skladu z Zakonom o DDV.

Posrednik prepovedan

Slika 2.8: Primer ročno izdelanega računa

1. če je blagajniška oprema implementirana kot **samostojen** modul, opisan proces olajša in pohitri le delno - uporabnik mora še zmeraj sam izračunati število nočitev, število posameznih turističnih taks in znesek dodatnih stroškov ter jih vnesti v blagajno (blagajna tako opravi le izračun skupnega zneska, DDV-ja in račun natisne),
2. če je blagajniški modul **povezljiv** z ostalimi moduli recepcijske programske opreme, lahko opisan proces avtomatizira do največje mere, vendar je v tem primeru potrebno v drugih moduli vnašati več podatkov ob kreaciji rezervacije in med nastankom dodatnih stroškov (kar predstavlja več dodatnega sprotne dela, vendar je nato možno račun generirati popolnoma samodejno).

Ne glede na nivo avtomatizacije uporaba blagajniškega modula pomeni veliko poenostavitev nadaljnje obdelave podatkov. Če je blagajniški modul dovolj napreden, lahko namreč sam generira tudi druge pomembne dokumente (blagajniški dnevnik, prejeme, izdatke, stanje salda in podobne). Naprednejše rešitve omogočajo tudi pripravo podatkov na način, ki poenostavi delo v računovodstvu.

Izbira rešitve je pogosto odvisna od velikosti nastanitvenega objekta. Ročna izdaja računov v velikih enotah ni smiselna, medtem ko je pri manjših še zmeraj zelo pogosta. Preprostejše samostojne blagajne, realizirane kot spletne aplikacije, so razmeroma cenovno ugodne in omogočajo delno avtomatizacijo. Funkcionalne povezljive blagajne, ki omogočajo popolno avtomatizacijo procesa, so za manjše objekte bodisi predrage ali pa so povezljive izključno z moduli istega proizvajalca.

2.2 Digitalizacija recepcijskega dela

S prejšnjim podpoglavjem je bralec pridobil osnovno predstavo o najnujnejših nalogah administrativnega recepcijskega dela in nekaj motivacije za uporabo recepcijske programske opreme. Brez namenske programske opreme so opi-

sane naloge lahko pogosto naporne in časovno zahtevne, tudi v najmanjših nastanitvenih objektih.

Ponovno je potrebno poudariti, da so naloge pri recepcijskem delu v manjšem in velikem objektu iste, vendar se bistveno razlikujejo v obsegu. Z drugimi besedami to pomeni, da je lahko implementacija elektronske knjige gostov za manjše objekte popolnoma drugačna od implementacije za velike objekte, a je elektronska knjiga gostov pri obeh potrebna. Množica modulov, ki je potrebna manjšim objektom, je tako enaka množici modulov, ki so v uporabi pri velikih, vendar je lahko nabor njihovih funkcionalnosti precej drugačen.

Razdelati je potrebno, kako je lahko recepcijska programska oprema zasnovana, kakšne so posledice izbire posameznega pristopa ter kateri med njimi so primerni za recepcijsko programsko opremo, ki je namenjena izključno manjšim nastanitvenim objektom.

Pri snovanju recepcijske programske opreme se mora razvijalec med drugim odločiti o:

- stopnji sklopljenosti modulov (šibka ali močna sklopljenost),
- načinu preprečevanja napak (striktno ali nestriktno preprečevanje),
- največjem številu sočasnih uporabnikov (enoinstančni ali večinstančni vmesnik),
- načinu hranjenja podatkov (lokalno ali strežniško) ter o
- načinu vzdrževanja programske opreme in obliki podpore (s servisno pogodbo ali brez).

2.2.1 Stopnja sklopljenosti modulov

Kot že rečeno, so moduli lahko samostojni ali povezljivi. Če so povezljivi, je lahko njihova medsebojna sklopljenost dosežena na dveh nivojih.

Pri *šibki sklopljenosti* modulov le-ti lahko delujejo kot samostojna orodja, ki ne zahtevajo uporabe ostalih. Kljub temu nudijo povezljivost, za katero

se mora uporabnik sam odločiti, ali jo bo koristil ali ne. Uporabnik lahko na ta način izbira, katere module zares potrebuje in drugih ne uporablja. Urejevalnik rezervacij, ki je šibko sklopljen s knjigo gostov, lahko tako recimo ob spremembi rezervacije sam popravi vpise v knjigi gostov, a če uporabnik urejevalnika ne uporablja, ima še zmeraj možnost vpise popraviti ročno.

Pri *močni sklopljenosti* modulov je uporaba vseh implementiranih modulov obvezna. Moduli delujejo kot celota in jih ni mogoče uporabljati kot samostojna orodja. Za vpis gosta, ki se v nastanitev prijavi brez rezervacije, je tako potrebno nujno najprej kreirati rezervacijo in določiti vse potrebne parametre, tudi če rezervacijskega modula uporabnik ne želi uporabljati.

Šibka sklopljenost tako pomeni pristop, pri katerem je način uporabe programske opreme odvisen od uporabnika samega, močna sklopljenost pa pristop, pri katerem je uporabnik prisiljen, da programsko opremo uporablja na natanko določen način. V praksi je težko govoriti o popolnoma šibko sklopljenih in močno sklopljenih implementacijah, saj je lahko sklopljenost različna tudi med posameznimi moduli. Običajno pa je implementacija recepcijske programske opreme kot celota pretežno šibko ali močno sklopljena.

Šibka sklopljenost se zaradi prilagodljivosti zdi boljši pristop od močne, vendar ima tudi slabosti, ki jih ne gre zanemariti:

- ker mora modul delovati kot samostojno orodje in hkrati biti povezljiv, je taka implementacija zahtevnejša in od razvijalca zahteva več truda in časa,
- možnost nastajanja napak je večja, zato je kvaliteta programske opreme kot celote zelo odvisna od tega, kako program te rešuje,
- uporabnik lahko določen modul najprej uporablja, nato z uporabo preneha, kasneje pa ga prične spet uporabljati, na kar morajo biti ostali moduli pripravljeni.

Med velikimi nastanitvenimi objekti je recepcijsko delo zelo podobno, saj se v velikih podjetjih uporabljajo že ustaljene, uveljavljene prakse, ki jih ni

enostavno spreminjati. Zaradi tega je splošen postopek dela v velikih objektih dobro definiran in ima zato veliko programskih rešitev, ki so namenjene največjim objektom, močno sklopljene module. Ta odločitev zahteva, da uporabnik programsko opremo dobro spozna, da se jo nauči uporabljati ter da jo uporablja na natanko določen način. V velikih objektih to ne predstavlja večjih težav, saj ponudniki take programske opreme večinoma nudijo tudi uvajanje.

Recepcijsko delo je med posameznimi manjšimi nastanitvami lahko zelo različno. Že nekaj enot več ali manj povsem spremeni obseg dela in posledično tudi njegov postopek. V nekaterih manjših objektih je na primer potrebno dnevno vpisovati po 50 novih gostov, pri drugih pa so enote oddane le nekajkrat letno. Potrebe se v takih objektih zelo razlikujejo, zato močna sklopljenost modulov ni primerna.

2.2.2 Preprečevanje napak

Zelo velik faktor, ki vpliva na enostavnost uporabe recepcijske programske opreme, je način preprečevanja napak. Pri implementaciji z močno sklopljenimi moduli je nevarnost napak veliko manjša kot pri uporabi šibko sklopljenih modulov, vendar je kljub temu prisotna.

Napaka je v recepcijski programski opremi lahko različnih oblik in pomeni:

- obstoj podatkov, ki ne ustrezajo resničnemu stanju (na primer obstoj vpisa gosta z nepravilnima datumoma odhoda in prihoda),
- neskladje podatkov med moduli (v knjigi gostov je na primer za določen dan vpisanih 10 gostov, urejevalnik rezervacij pa jih v poročilu navede le 8) ali
- obstoj podatkov, ki jih en modul interpretira kot ustrezne, drugi pa kot neustrezne.

Dobra recepcijska programska oprema mora minimizirati ali povsem preprečiti nastajanje napak, obstoječe napake pa mora prepoznati in nanje upo-

rabnika opozoriti. Pri snovanju modulov lahko razvijalec izbira med dvema pristopoma, in sicer med striktnim preprečevanjem napak ali nestriktnim.

Pri *striktnem preprečevanju napak* programska oprema ne dovoli vnosa podatkov, ki bi povzročili napako. Pri *nestriktnem preprečevanju napak* program vnosa nekaterih napak ne dovoljuje, medtem ko vnos drugih dovoljuje, vendar nanje uporabnika med delom opozarja.

Izbira pristopa zelo vpliva na uporabniško izkušnjo in, podobno kot pri izbiri stopnje sklopljenosti modulov, odloča o tem, kako enostavna je programska oprema za uporabo. Striktno preprečevanje napak je zagotovo dober pristop, saj je lahko uporabnik manj pozoren pri vnosu in urejanju podatkov. Vendar pa tako preprečevanje pomeni, da mora razvijalec programske opreme predvideti vse nevarnosti, kjer do napak lahko pride. Če je kaka izmed njih pozabljena, lahko uporabnik v prepričanju, da do napak ne more priti, v sistem vnese napako, ki jo je kasneje težko odkriti. Poleg tega mora programska oprema vsebovati veliko več orodij za reševanje situacij, ki niso lahko predvidljive.

Denimo, da knjiga gostov s striktnim preprečevanjem napak ne dovoli vpisa gosta z datumom odhoda, ki je krajši od datuma konca rezervacije, kateri gost pripada. Razvijalcu programske opreme se morda to zdi popolnoma v redu, saj ni razloga, da bi bila rezervacija daljša od dejanskega bivanja njenih gostov, in zato poskus vpisa razume kot napako. V praksi ni tako in obstajajo situacije, ko je taka napaka povsem utemeljena - gost lahko na primer nastanitev zapusti prej kot ostali gostje, s katerimi je opravil rezervacijo in bival v isti enoti. Uporabnik programa, pri razvoju katerega razvijalci takega primera niso predvideli, je tako primoran ugotoviti, kako tako situacijo razrešiti. To običajno pomeni, da mora uporabnik poseči po orodjih, ki v osnovi niso namenjena razreševanju natanko te situacije. V danem primeru bi bil lahko uporabnik primoran celotno rezervacijo preklicati (stornirati) in jo nadomestiti z dvema novima, pri čemer bi ena zajemala obdobje, v katerem so bili vsi gostje v nastanitvi, druga pa obdobje, v katerem je bil eden izmed njih odsoten.

Opisan primer lepo nakazuje, da je razvoj programske opreme s striktnim preprečevanjem napak lahko izjemno zahteven in da je kvaliteta take opreme odvisna od števila in vrste izjemnih situacij, ki jih je skupina razvijalcev predvidela. Poleg tega morajo biti uporabniki take programske opreme zelo dobro seznanjeni z vsemi orodji, ki jih oprema ponuja, kar zahteva uvajanje in izkušnje.

Na drugi strani pa lahko recepcijska programska oprema z nestriktnim preprečevanjem napak ob poskusu vpisa gosta, ki ima datum odhoda krajši od datuma zaključka rezervacije, uporabnika le obvesti, da poskuša storiti nekaj, kar ni običajno in kar lahko pomeni napako. Uporabnik ve, da ne gre za napako, temveč za poseben primer, in potrdi, da želi vpis kljub temu dodati. Uporabnik si lahko na poljuben način zabeleži, zakaj je potencialno napako dovolil, in brez ovir nadaljuje z delom. Dober program bo omogočal tudi kasnejši pregled ali iskanje obstoječih napak. V primeru, da uporabnik napak v svojem sistemu ne želi imeti, ima še zmeraj možnost vpisovanje prekiniti in poiskati rešitev z uporabo drugih orodij.

Pri velikih nastanitvenih objektih z recepcijsko programsko opremo upravlja razmeroma veliko število ljudi, zato je vnos napak potrebno povsem preprečiti - tudi takih, ki v resnici predstavljajo le neobičajen primer in ne resnične napake. V velikih objektih je zato večja potreba po striktnosti preprečevanja napak kot pa po enostavnosti uporabe programske opreme.

Pri manjših nastanitvah je ravno obratno. Z recepcijsko programsko opremo običajno upravlja manjše število ljudi, ki od nje pričakujejo večjo prilagodljivost in možnost lastnega načina reševanja nepogostih situacij. Pri manjšem številu uporabnikov je vodenje evidence posebnih situacij enostavnejše in je napake kasneje enostavno najti in odpraviti, če je to potrebno. Pri manjših nastanitvah je zato nestriktnost preprečevanja napak dobrodošla.

2.2.3 Število sočasnih uporabnikov

Število uporabnikov, ki lahko z recepcijsko programsko opremo dela sočasno, je pogojeno z arhitekturo rešitve. Če želi programska oprema omogočati

sočasno delo na več različnih napravah, mora biti uporabniški vmesnik zasnovan *večinstančno*. Podatkovno bazo je potrebno ločiti od uporabniških vmesnikov in jo namestiti na napravo, do katere lahko dostopajo vsi uporabniki. Razviti je potrebno vmesno programsko opremo (*middleware*), ki skrbi za sinhronizacijo med instancami uporabniškega vmesnika in preprečuje sočasno spreminjanje istih podatkov. Razvoj take rešitve je veliko zahtevnejši v primerjavi z razvojem *enoinstančne* rešitve.

Pri velikih nastanitvah uporaba enoinstančnih rešitev zagotovo ni primerna. V recepcijah takih objektov rezervacije običajno sočasno upravljata najmanj dve osebi.

Pri manjših nastanitvah je večinstančnost recepcijske programske opreme zagotovo dobrodošla, vendar v večini primerov ni kritičnega pomena. Poleg tega si veliko manjših nastanitvenih enot ne more privoščiti drage dodatne strojne opreme, ki jo zahteva večinstančna arhitektura. Pri takih objektih se v recepciji običajno nahaja en sam računalnik, na katerem se izvajajo vsa računalniško podprta administrativna dela.

Seveda ima razvijalec recepcijske programske opreme možnost svojo rešitev implementirati kot storitev v oblaku, pri čemer strošek nabave (ali najema) potrebne strojne opreme prenese nase. V tem primeru pa mora dobro zastaviti način obračunavanja svojih storitev in od uporabnikov zahtevati redna plačila za uporabo.

2.2.4 Hranjenje podatkov

Hranjenje podatkov je lahko lokalno ali strežniško. Pri *lokalnem hranjenju podatkov* so vsi podatki, s katerimi recepcijska programska oprema dela, na napravi, na kateri je ta oprema nameščena. Podatkovna baza je v tem primeru integrirana v samo programsko opremo. Pri *strežniškem hranjenju podatkov* se podatkovna baza lahko nahaja na drugi napravi, do katere mora imeti recepcijska programska oprema dostop.

Način hranjenja podatkov je tesno povezan z odločitvijo o največjem številu sočasnih uporabnikov, ki ga rešitev omogoča. Pri večinstančnem pri-

stopu je strežniško hranjenje podatkov nujno. To pa ne pomeni, da mora biti hranjenje podatkov pri enoinstančnih rešitvah lokalno.

Podatkovna baza je pri enoinstančnih rešitvah še zmeraj lahko ločena komponenta opreme. Razvijalec se za tako rešitev lahko odloči, če želi le podatkovno bazo ponuditi kot storitev v oblaku, ostale recepcijske programske opreme pa ne. Drug razlog za izbiro take rešitve pa bi lahko bila večja stopnja varnosti v primeru okvare naprave v recepciji. Ker so te naprave pri nekoliko večjih nastanitvah neprestano v uporabi, so posledično tudi bolj izpostavljene morebitnim okvaram. Premik podatkovne baze na ločen strežnik je tako dobrodošla nadgradnja, vendar se je potrebno zavedati, da se okvara lahko zgodi tudi v njem. V vsakem primeru je redna izdelava varnostnih kopij podatkovne baze nujno opravilo, ki ga je potrebno izvajati tudi pri rešitvah z lokalnim hranjenjem podatkov.

Če se razvijalec zavoljo lažjega in hitrejšega razvoja pri zasnovi odloči za enoinstančni pristop, posebne motivacije za implementacijo hranjenja podatkovne baze na strežniški način nima.

2.2.5 Vzdrževanje programske opreme in podpora

Del stroškov, ki jih zaradi uporabe recepcijske programske opreme imajo veliki nastanitveni objekti, predstavlja vzdrževanje. To je običajno izvedeno v obliki *servisne pogodbe*, ki jo podpišeta razvijalec in nastanitev. Razvijalec se s tako pogodbo v zameno za letno plačilo obveže, da bo nastanitvi v primeru težav ali nejasnosti pri uporabi njihove programske opreme na voljo in da bo morebitne težave odpravil. V servisno pogodbo se med drugim tudi zapiše, za katere morebitne nepravilnosti odgovarja razvijalec in za katere uporabnik. Pri velikih nastanitvenih objektih, kjer se posluje z velikimi količinami sredstev, je sklenitev take pogodbe nujna.

Pri manjših nastanitvah je servisna pogodba lahko dobrodošla, vendar zaradi dodatnih stroškov, ki jih prinaša, velikokrat ni. Pogosteje se servisne storitve izvajajo na zahtevo, pri čemer razvijalec zaračuna delo, ki ga opravi. Če si nastanitev zagotovi recepcijsko programsko opremo, s katero pri nadalj-

nji uporabi nima težav ali je ne želi nadgraditi, lahko strošek digitalizacije in avtomatizacije recepcijskega dela minimizira.

Poglavje 3

Pregled obstoječih rešitev na slovenskem trgu

V prejšnjem poglavju so bili opisani pristopi, med katerimi razvijalec recepcijske programske opreme lahko izbira. Pri opisu posameznega pristopa so bile identificirane potrebe manjših nastanitvenih ponudnikov, ki so naslednje:

- šibka sklopljenost modulov,
- nestriktno preprečevanje napak,
- enoinstančna ali večinstančna zasnova,
- lokalno hranjenje podatkovne baze in
- vzdrževanje ter podpora brez servisne pogodbe.

Izbira naštetih pristopov pa še ne zagotavlja, da bo ciljni uporabnik zadovoljen tudi s samo implementacijo. Uporabniki v manjših nastanitvenih objektih namreč iščejo take rešitve, ki:

- so enostavne za namestitev brez pomoči računalniških strokovnjakov,
- so enostavne za uporabo in je njihova uporaba samoumevna,
- recepcijsko delo poenostavijo in pohitrijo,

- omogočajo reševanje nepredvidljivih situacij ter
- zmanjšujejo dodatne stroške.

Upoštevajoč dejstvo, da manjši nastanitveni ponudniki v Sloveniji predstavljajo več kot 2 tretjini vseh nastanitvenih objektov, bi bilo pričakovati, da je na slovenskem trgu izbira recepcijske programske opreme zanje pestra. Raziskava obstoječih rešitev pokaže, da temu ni tako. Različnih rešitev je malo, tudi te pa imajo pogosto enake pomanjkljivosti.

V nadaljevanju sledijo opisi štirih najbolj pogosto uporabljenih obstoječih rešitev, ki skorajda predstavljajo celotno ponudbo za manjše nastanitve na slovenskem trgu. Poleg štirih opisanih rešitev je moč najti še kakšno, vendar je njihova uporaba zelo redka ali pa gre za rešitve, ki so jih nastanitve razvile za lastne potrebe.

Preizkus uporabe in pregled delovanja je zajemal programe Recepcija Polis in Birokrat Hotelir ter spletni aplikaciji iGost in ProGOST.

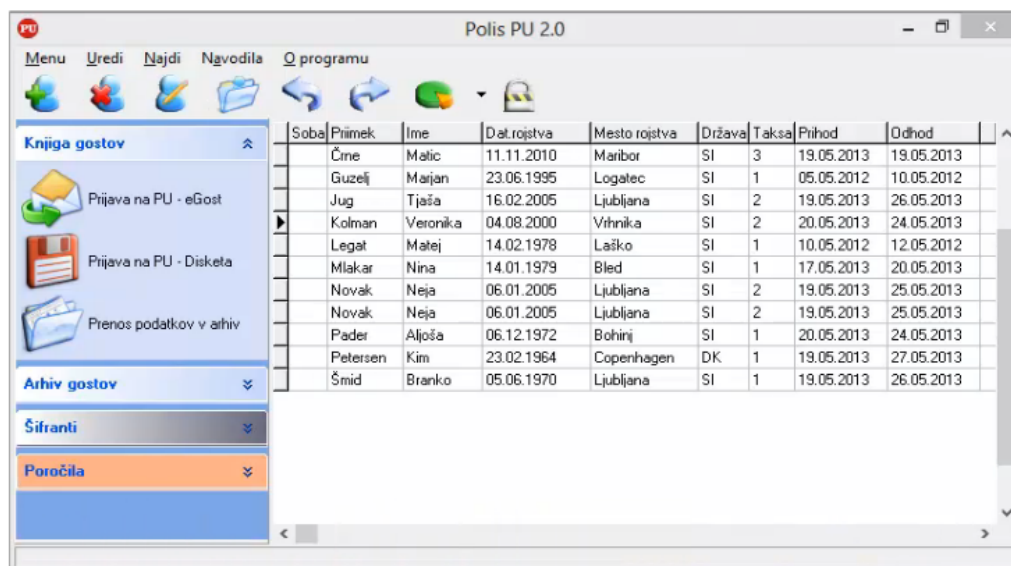
3.1 Recepcija Polis

Recepcijska programska oprema, ki jo na trgu ponuja DB Studio z Bleda kot program *Recepcija Polis*, je manjšim nastanitvam na voljo v dveh različicah:

- Polis Pu2 ali
- Polis 3.

Različica Polis Pu2 je zasnovana kot samostojen, nepovezljiv modul, ki implementira le knjigo gostov. Omogoča vpis gostov, njihovo evidentiranje ter prijavo na policijo preko interneta. Poleg tega vsebuje orodji za izračun števila prihodov in nočitev v poljubnem časovnem obdobju ter izdelavo statističnega poročila po narodnosti gostov.

Uporabniški vmesnik različice Pu2 je preprost in vpisane goste prikaže v obliki tabele, kot je moč videti na sliki 3.1.



Slika 3.1: Uporabniški vmesnik programa Polis Pu2

Glavni prednosti te rešitve sta preprostost in cenovna ugodnost. Strošek predstavlja le nakup licence, saj obvezne servisne pogodbe ni. Cena je odvisna od števila vpisanih gostov z enakim datumom prihoda in znaša:

- 100 evrov, če je največje dovoljeno število vpisanih gostov z enakim datumom prihoda 10,
- 150 evrov, če je največje dovoljeno število vpisanih gostov z enakim datumom prihoda 20 in
- 200 evrov, če je največje dovoljeno število vpisanih gostov neomejeno.

Različica Pu2 je zaradi enkratnega plačila med najugodnejšimi rešitvami, vendar le ob pogoju, da je rešitev v uporabi vsaj nekaj let.

Pri preizkusu delovanja programa pa se izkaže, da ima programska oprema Polis Pu2 nekaj pomembnih slabih lastnosti, ki se pri drugih rešitvah ne pojavljajo:

- filtriranje seznama gostov ni mogoče (v tabeli so zmeraj prikazani vsi vpisi, ki so bili kadarkoli vnešeni, razen tistih, ki so premaknjeni v arhiv, kar zelo zmanjša preglednost),
- premikanje po tabeli z vrtenjem miškega kolesčka ne deluje (pri daljši uporabi programa postane namreč tabela zelo velika in je premikanje po njej neprijazno opravilo),
- operacije nad podatki so pri večjem številu gostov izredno počasne (izdelava prijave za policijo, poročila prihodov in nočitev ali statistike lahko pri nekoliko večjem številu vpisov traja tudi deset minut ali več¹) in najpomembneje,
- preprečevanje napak ni implementirano.

Pri zadnji trditvi se je potrebno ustaviti in preveriti, kje je pomanjkljivost opazna ter kakšne so njene posledice. Uporabnik programa podatke vnaša s pomočjo obrazca, ki je prikazan na sliki 3.2.

Elektronske knjige gostov lahko načeloma dovoljujejo vpis poljubne množice znakov in vrednosti, vendar morajo pri prijavi teh podatkov na policijo podatke obdelati tako, da so skladni s pravili programskega paketa eGost. Oddaja podatkov v nepravilni obliki je prepovedana in je lahko kaznovana z oglobitvijo nastanitve (več o tem v poglavju 4.2).

Obrazec za vnos podatkov programa Polis Pu2 dovoljuje vnos poljubnih znakov in vrednosti ne glede na to, ali so smiselni ali ne. Tako je na primer

¹Podrobna analiza pokaže, da je za to slabost kriva časovna zahtevnost algoritma, ki ga program uporablja za obdelavo vpisov. Pri izdelavi statistike števila nočitev in prihodov po državah za določeno obdobje algoritem najprej preveri vsako vrstico v tabeli in pri tistih, kjer datuma prihoda in odhoda ustrezata izbranemu časovnemu okviru, prebere državo in jo doda v množico ciljnih držav. V naslednjem koraku algoritem za vsak dan v izbranem obdobju in za vsako državo iz množice vsako vrstico v tabeli zopet obišče in presodi, ali je potrebno število nočitev in prihodov pri državi iteracije in dnevu iteracije inkrementirati. Časovna zahtevnost je tako odvisna od števila vseh vrstic v tabeli, pri čemer jih večina sploh ni aktualnih, dolžine časovnega okvira ter števila različnih držav vpisov.

Splošni podatki gostov

Priimek

Ime

Datum rojstva

Kraj rojstva

Država

Taksa

Prihod

Odhod

EMŠO/Pas.No.

Soba

Opomba

Novi Gost Potrdi Izhod

Slika 3.2: Obrazec za vnos podatkov rešitve Polis Pu2

The screenshot shows a software window titled "Splošni podatki gostov" (General guest data) with a red "PU" logo in the top-left corner. The window contains several input fields and dropdown menus. The "Datum rojstva" (Date of birth) field is filled with "99.99.9999". The "Prihod" (Arrival) and "Odhod" (Departure) fields are filled with "24. 4 .2013". At the bottom of the window, there are three buttons: "Novi Gost", "Potrdi", and "Izhod".

Slika 3.3: Obrazec za vnos podatkov rešitve Polis Pu2 z vnešenimi vrednostmi, ki niso smiselne

v polje za vnos priimka mogoče vnašati ločila ter v polje za vnos rojstnega datuma vpisati vrednosti, ki ne predstavljajo realnih datumov, kot je to mogoče videti na sliki 3.3.

Pri pošiljanju prijav gostov na policijo Polis Pu2 podatkov ne obdela in jih pošlje takšne, kakršne je uporabnik vpisal. Posledično mora uporabnik sam zelo paziti, da so podatki, ki jih vpisuje, skladni s pravilnikom paketa eGost ter pravila dobro poznati. V primeru, da se uporabnik zatipka in napake ne opazi, program podatke sprejme brez opozorila. Nevarnost za oddajo neustreznih podatkov na policijo je tako velika, posledice pa so lahko za manjšo nastanitev velikanske.

Kljub preprostosti uporabniškega vmesnika je tako rešitev Polis Pu2 z vidika prijaznosti uporabe in varnosti neprimerna.

	10.12	11.12	12.12	13.12	14.12	15.12	16.12	17.12	18.12	19.12	20.12
dvop											
dvop											
dvop											
dvop											
dvop											
dvop											
dvop											
dvop											
dvop		000033	000033	000033	000033	000033	000033	000033	000033	000033	000033
0 dvop		000033	000033	000033	000033	000033	000033	000033	000033	000033	000033
1 TR0											
2 TR0		xx	xx	xx	xx	xx	xx	xx	xx	xx	xx
3 TR0											
4 TR0											
5 TR0		000034	000034	000034	000034	000034	000034	000034	000034	000034	000034
6 TR0											

Slika 3.4: Prikaz zasedenosti sob (mreža) pri rešitvi Polis 3

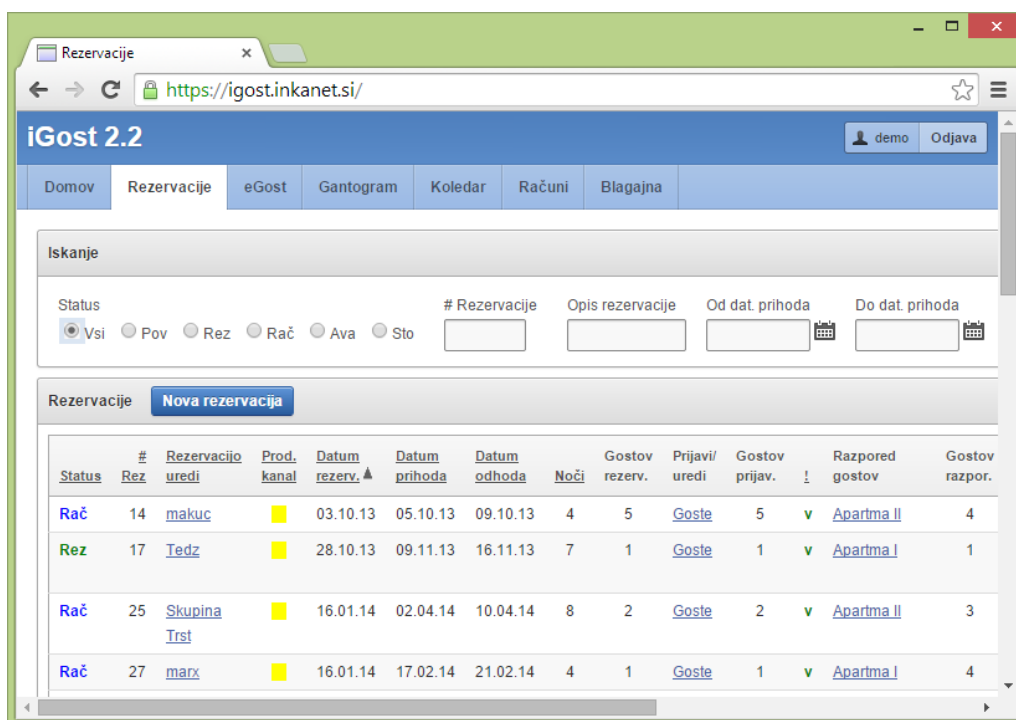
Preizkus rešitve Polis 3, ki poleg knjige gostov nudi tudi urejevalnik rezervacij in blagajno, ni bil mogoč. Iz prosto dostopnih navodil za uporabo pa je mogoče ugotoviti, da uporaba programa ni samoumevna ter da uporabniški vmesnik ni vizualno sodoben in da je slabo pregleden. Na sliki 3.4 je kot primer prikazan pregledovalnik zasedenosti sob. Mreža ni interaktivna, na njej so rezervacije označene le s številkami, dodatnih vizualnih informacij ni.

Različica Polis 3 je v primerjavi z različico Pu2 tudi veliko dražja. Strošek nakupa opreme znaša 1000 evrov, poleg tega pa je obvezno tudi letno vzdrževanje, ki znaša najmanj 250 evrov letno.

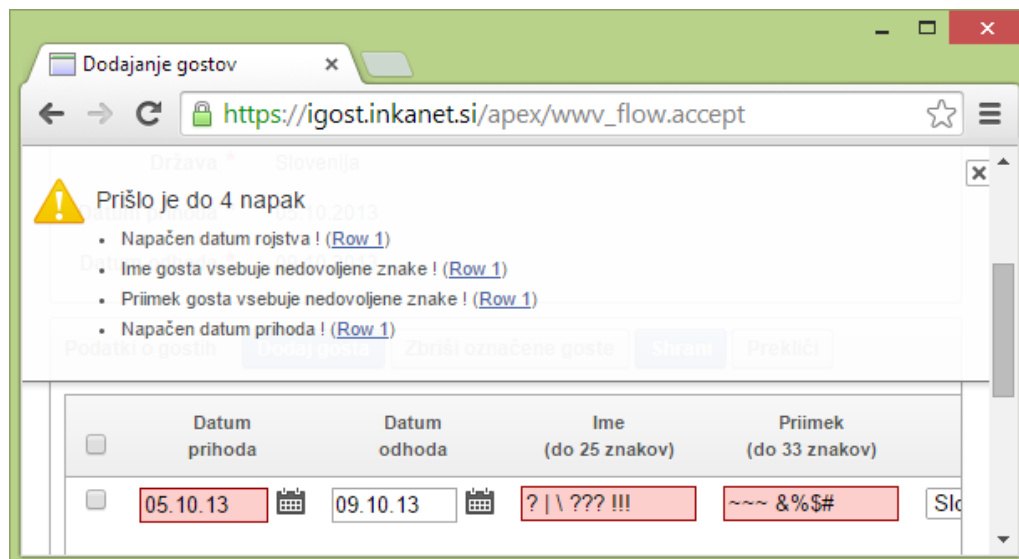
3.2 iGost

iGost je sodobnejša, vizualno prijaznejša in popolnoma drugače zastavljena rešitev, ki jo podjetje Inkanet ponuja v obliki storitve v oblaku. Implementirana je kot spletna aplikacija (slika 3.5) in za delovanje potrebuje internetno povezavo in brskalnik. Vključuje knjigo gostov, urejevalnik rezervacij, gotovinsko blagajno in množico dodatnih funkcij za poenostavitev dela.

iGost ponuja tudi možnost testne uporabe pred nakupom, kar je omogočilo



Slika 3.5: Uporabniški vmesnik rešitve iGost



Slika 3.6: Striktno preprečevanje napak v programu iGost ne dovoli vpisa podatkov, ki niso skladni s pravili programskega paketa eGost

dober preizkus njegovega delovanja. Zaradi pomanjkanja preprečevanja napak pri rešitvi Polis Pu2, je bila to ena izmed prvih funkcionalnosti, ki je bila preverjena. Aplikacija v celoti napake preprečuje striktno, kar je pri vnosu osebnih podatkov gostov dobrodošlo. Pri ponovitvi poskusa vnosa nesmiselnih podatkov je iGost napake prepoznal in preprečil njihov vnos, kot je to mogoče videti na sliki 3.6.

Urejevalnik rezervacij nudi tudi mrežo, ki je veliko preglednejša od tiste v programu Polis 3. Mreža je prikazana na sliki 3.7. Ob premiku miške nad posamezno rezervacijo se v obliki manjšega lebdečega okna (*tooltip*) prikažejo ime rezervacije, datum prihoda in datum odhoda. Klik na rezervacijo prikaže dodatne informacije nad mrežo in ponudi njeno urejanje.

Kljub opisanima bistvenima izboljšavama v primerjavi z rešitvijo Polis 3 ima iGost nekaj lastnosti, ki so lahko pri manjših nastanitvenih enotah nezaželene in moteče:

- moduli so močno sklopljeni, kar onemogoča samostojno uporabo posa-

The screenshot shows the iGost reservation interface. At the top, there is a date selection field with '09.01.15' and a calendar icon, followed by an 'Osveži' (Refresh) button. To the right, it displays 'Rezervacija: test od 19.01.15 do 25.01.15 gostov:2' and a 'Uredi' (Edit) link. Below this is a calendar for 'Januar 2015' with days 12 to 24. The days of the week are abbreviated as P, T, S, Č, P, S, N, P, T, S, Č, P, S. A table below the calendar shows reservations for 'Povpraševanje' (0 oseb), 'Apartma I' (2 oseb), and 'Apartma II' (3 oseb). The 'Povpraševanje' row is marked 'zaprto' (closed) for the first 18 days. The 'Apartma I' row has an orange 'test' button on the 19th. The 'Apartma II' row is empty. At the bottom, there are navigation controls including '1 of 1' and various arrow buttons.

Slika 3.7: Prikaz rezervacij v mreži pri rešitvi iGost

meznega modula (če želi uporabnik le gosta prijaviti na policijo, mora nujno najprej zanj ustvariti rezervacijo),

- preprečevanje napak ni striktno le pri vnosu osebnih podatkov, ampak v celotni aplikaciji (določene nepredvidene situacije je zelo težko rešiti),
- integracija s paketom eGost ni avtomatizirana (aplikacija pripravi datoteko **PRIJAVN.DAT**, ki jo mora uporabnik shraniti na svoj računalnik, premakniti v ustrezen direktorij in ročno pognati eGost),
- ponovno pošiljanje prijav na policijo ni mogoče,
- popraviljanje podatkov že vpisanih in na policijo prijavljenih podatkov ni mogoče ter
- nekateri deli uporabniškega vmesnika so nepregledni in hroščati.

Obračunavanje uporabe recepcijske programske opreme iGost poteka na letni ravni in je možno v obliki dveh paketov, med katerima lahko uporabniki izbirajo:

- *Maxi paket* ne vključuje gotovinske blagajne in prikaza rezervacij v mreži, je pa zato nekoliko cenejši (48 evrov letno za največ 20 ležišč in dodatnih 12 evrov letno za vsakih nadaljnjih 20 ležišč), medtem ko
- *paket Pro* vključuje vse module, vendar je cenovno manj ugoden (72 evrov letno za največ 20 ležišč in dodatnih 18 evrov letno za vsakih nadaljnjih 20 ležišč).

3.3 ProGOST

ProGOST, pogosta izbira mnogih sobodajalcev, je prav tako implementiran kot storitev v oblaku. Podobno kot iGost je ProGOST v obliki spletne aplikacije, do katere uporabnik dostopa z brskalnikom. Aplikacijo ponuja podjetje PRONET iz Kranja kot eno izmed svojih mnogih rešitev za podporo poslovanja.

Aplikacija implementira le knjigo gostov, ob doplačilu pa nudi integracijo z ločeno aplikacijo ProGRAM Fakturiranje, enim izmed ostalih produktov podjetja.

Največja pomanjkljivost te rešitve je odsotnost urejevalnika rezervacij. Aplikacija je tako po naboru funkcionalnosti zelo podobna rešitvi Polis Pu2. Implementirane so le najnujnejše, kot so integracija s paketom eGost, izdelava poročil za občino in Statistični urad, drugih orodij, s katerimi bi si uporabnik lahko olajšal delo, ne ponuja, z izjemo:

- možnosti kopiranja podatkov že vpisanega gosta (pohitri vpis gosta, ki je nekoč že bival v nastanitvenem objektu) in
- avtomatizirane integracije s paketom eGost (uporabniku ni potrebno datoteke PRIJAVN.DAT ročno nameščati v ustrezen direktorij in ročno sprožati pošiljanja na policijo).

Zaračunavanje uporabe te rešitve je osnovano na številu gostov, ki jih uporabnik v aplikacijo vpiše, in fiksnem mesečnem pavšalu. Za uporabo

polnega nabora funkcij, ki so nujne za manjši namestitveni objekt, znaša mesečni pavšal 68 evrov, dodatni strošek za vsak vpis gosta pa 0,05 evra. Za nadgradnjo rešitve z gotovinsko blagajno je potrebno mesečno odšteti dodatnih 12 evrov.

Glede na to, da rešitev ProGOST v primerjavi z že opisanima rešitvama nima bistvenih prednosti, predstavlja eno izmed najdražjih rešitev.

3.4 Birokrat Hotelir

Celovita recepcijska programska oprema Birokrat Hotelir podjetja Birokrat je prav tako ena izmed najpogostejše izbranih rešitev, vendar navadno pri nekoliko večjih nastanitvenih objektih, saj je v primerjavi z ostalimi rešitvami veliko dražja. Program odlikujeta širok spekter dodatnih funkcionalnosti, ki niso nujne, ampak lahko bistveno olajšajo recepcijsko delo, ter povezljivost z ostalimi produkti podjetja. Primer take funkcionalnosti je interaktivnost grafičnega prikaza urejevalnika rezervacij, ki omogoča, da uporabnik rezervacijo doda z risanjem termina rezervacije neposredno v mrežo.

Rešitev Hotelir poskuša biti sočasno primerna tako manjšim kot večjim nastanitvenim objektom, kar ni najboljša izbira. Večji nastanitveni objekti zahtevajo močno sklopljenost modulov ter striktno preprečevanje napak, za kar je bralec že izvedel, da močno zmanjša prijaznost in enostavnost uporabe. Preizkus demonstracijske različice programa, ki je na voljo na spletni strani podjetja, to trditev tudi potrdi. Uporabniški vmesnik je poln najrazličnejših funkcij, vendar vse prej kot prijazen za uporabo [4]. Primer vnosne maske za dodajanje sobe je prikazan na sliki 3.8.

Pregled uradnega cenika razkrije, da redna cena samo knjige gostov (brez drugih modulov in z najosnovnejšimi funkcijami) znaša 399 evrov, pri čemer je za poln nabor funkcij potrebno odšteti najmanj 589 evrov. Ker uporaba programa ni samoumevna, je v ceno vključen tudi uvajalni tečaj.

Zaradi visoke cene, zahtevnosti uporabe in neprijaznosti uporabniškega vmesnika je ta rešitev za manjše nastanitvene objekte neprimerna, za nekoliko

Slika 3.8: Vnosna maska za dodajanje sobe s programom Birokrat Hotelir

večje pa le delno.

3.5 Pomanjkljivosti predstavljenih rešitev

Funkcionalnosti, načini implementacije in cena (ter načini zaračunavanja) so pri štirih predstavljenih rešitvah zelo različni. Primernost rešitev za uporabo v manjših nastanitvenih objektih je pri rešitvah iGost in ProGOST največja. Rešitev Polis zaradi manjkajočega preprečevanja napak ni primerna za uporabo. Rešitev Birokrat Hotelir pa je za večino manjših nastanitev prekompleksna, da bi bilo z njim mogoče recepcijsko delo pohitrili in poenostaviti.

Vse izmed opisanih rešitev, ne glede na način implementacije, imajo velike pomanjkljivosti v naboru dodatnih funkcionalnosti za resnično pohitritev, poenostavitev in izboljšavo recepcijskega dela v manjših nastanitvenih objektih. Ravno te funkcionalnosti so tiste, ki jih uporabniki najpogosteje pogrešajo. Med najbolj iskane sodijo:

- popolna integracija s paketom eGost z avtomatiziranim nadzorom ce-

lotnega procesa,

- samodejna priprava novih prijavnic ob spremembi podatkov gostov,
- hitro dodajanje, pregledovanje in urejanje rezervacij na grafičen način z operacijami nad mrežo, ki je čimbolj podobna klasičnim, fizičnim mrežam,
- orodja za hitro ugotavljanje, ali je v določenem terminu želena enota razpoložljiva,
- orodja za urejanje obstoječih rezervacij, s katerimi je mogoče optimizirati razpoložljivost,
- orodja za enostavno krajsanje in podaljševanje bivanja gostov ter zamenjavo enote (selitev),
- prepoznava nepotrjenih rezervacij s preteklim rokom za potrditev ter
- bogata poročila za pregled stanja enot, razpoložljivosti in dnevnih dogodkov.

Poleg omenjenih funkcionalnosti nobena izmed opisanih rešitev ne vključuje upravljanja z zunanjimi kanali.

Razlog za pomanjkanje naštetih funkcionalnosti najverjetneje tiči v tem, da razvijalci programske opreme recepcijsko delo spoznajo le površinsko in mu niso izpostavljeni dovolj dolgo, da bi utegnili prepoznati, katere manjkajoče funkcionalnosti, ki so sicer enostavne za implementacijo, lahko odločajo o tem, ali program recepcijsko delo olajšuje ali ga celo omejuje in otežuje.

Razvoj recepcijske programske opreme za manjše nastanitve zahteva tesno sodelovanje z uporabniki, zato mora razvijalec svojo rešitev venomer posodabljalati in pri tem upoštevati vsa mnenja uporabnikov, tudi če se mu morda zdijo predlagani popravki in nadgradnje nenujni.

Ker na slovenskem trgu rešitev za manjše nastanitve, ki bi bile razvite in vzdrževane s temi načeli, ni (ali pa so občutno predrage), je taka programska oprema zelo iskana in predstavlja tržno nišo.

Poglavje 4

Lastna rešitev

Prejšnji poglavji opisujeta potek recepcijskega dela v manjših nastanitvenih enotah ter obstoječe računalniške rešitve, namenjene digitalizaciji in avtomatizaciji tega procesa. Čeprav take rešitve na slovenskem trgu že obstajajo, so bodisi funkcionalno pomanjkljive, implementacijsko neustrezne ali pa za običajnega uporabnika predrage.

Osrednji predmet diplomske naloge sta razvoj in implementacija lastne rešitve, ki bi zadovoljila doslej neuslišane želje uporabnikov. Zasnova programa in množica njegovih funkcionalnosti temeljita na lastnih izkušnjah iz dela v recepciji srednje velikega hotela, kjer se je ideja za projekt tudi porodila.

4.1 Zasnova programa in načrt razvoja

Pri zasnovi lastne rešitve so kot glavna načela uporabljena tista, ki so na začetku poglavja 3 predstavljena kot najprimernejša za recepcijsko programsko opremo za manjše nastanitvene objekte.

Lastna rešitev je tako zasnovana modularno, pri čemer so moduli šibko sklopljeni. Velik poudarek je na preprečevanju napak, ki je nestriktne narave, z izjemo pri vnosu podatkov, ki se pošiljajo na policijo, kjer je preprečevanje napak striktno. Z namenom poenostavitve problemske domene je program

zasnovan enoinstančno, kar je za večino ciljnih uporabnikov povsem zadovoljivo. Želja po lokalnem hranjenju podatkovne baze je poglavitni razlog, da rešitev ni zasnovana kot storitev v oblaku, ampak namestljiv program. Namestitev je enostavna in jo lahko izvede uporabnik sam.

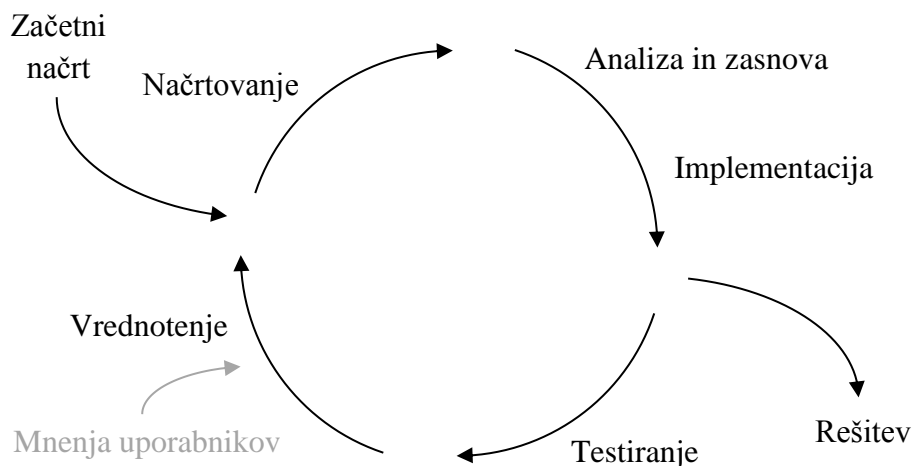
Uporabniški vmesnik je tisti, ki lahko ne glede na kvaliteto ostalih delov programa uporabnika takoj pritegne ali odvrne. Avtor Wilbert O. Galitz v svoji odlični knjigi [5] celo zatrjuje, da je uporabniški vmesnik najpomembnejši del vsake programske opreme in da strogo odloča o tem, ali bo produkt na trgu uspešen ali ne. Rešitve, opisane v poglavju 3, prav tako pa tudi rešitve, ki jih uporabljajo največji hoteli, posvečajo uporabniškemu vmesniku premalo pozornosti. Ti so pogosto vizualno neprivlačni ali preplavljeni s kontrolami, zaradi katerih se uporabniku delo upira, saj uporaba programa ni samoumevna in mora veliko časa porabiti, da se ga nauči uporabljati. Uporabniški vmesnik lastne rešitve je tako zasnovan kot poglavitna komponenta, vizualno zgledovana po vmesnikih najpopularnejših računalniških programov, s katerimi uporabniki radi delajo in jih dobro poznajo.

Z željo po čimvečjem številu končnih uporabnikov in hitri popularizaciji je lastna rešitev zasnovana kot brezplačna programska oprema, zgrajena na brezplačni (delno odprtokodni) platformi in iz odprtokodnih komponent.

4.1.1 Iterativno inkrementalni in uporabniško usmerjen razvoj

V podpoglavju 3.5 je bralec lahko ugotovil, da razvoj kvalitetne recepcijske programske opreme ne more biti izveden v enem samem koraku, vendar se poglavitni del razvoja prične šele z uporabo rešitve v praksi. Odziv uporabnikov je bistvenega pomena, saj razvijalec programske opreme nikakor ne more predvideti popolnoma vseh funkcionalnosti, ki so za uporabnika nujne. Uporabnike je potrebno vključiti v razvojni proces in jih venomer spodbujati, da izrazijo svoja mnenja, opazanja in ideje.

Iterativno inkrementalni pristop [6] k razvoju programske opreme nikakor ni nova oblika razvojnega modela. Kot uporabno tehniko pri razvoju pro-



Slika 4.1: Faze razvoja programske opreme pri iterativno inkrementalnem razvoju. Vključitev uporabnikov v razvojni proces je mogoč v fazi vrednotenja.

gramske opreme jo je Victor R. Basili predlagal že leta 1975, ko je poznavanje problematike do potankosti veljalo za predpogoj razvoja dobre programske opreme [7].

Na sliki 4.1 je grafični prikaz faz iterativno inkrementalnega pristopa. V vsaki iteraciji razvojnega cikla razvijalec svojo rešitev ponovno testira in vrednoti, načrtuje potrebne nadgradnje, analizira njihov učinek, se odloči, kako bodo le-te uvedene, in jih nazadnje implementira. S tem uporabnikom dostavi novo rešitev, ki je funkcionalna nadgradnja prejšnje. Z novo iteracijo se pričneta novi fazi testiranja in vrednotenja. Narava takega razvojnega procesa je idealna za vključitev uporabnikov v sam razvoj. Ob izdaji nove verzije rešitve se namreč faza testiranja ne prične le pri razvijalcu, pač pa tudi pri uporabnikih. V fazi vrednotenja ima tako razvijalec možnost poleg svojih rezultatov testiranja upoštevati tudi mnenja uporabnikov.

Iterativno inkrementalni razvoj tako omogoča vključitev uporabnikov, vendar ga sam po sebi ne zahteva niti spodbuja. Razvijalec mora sam razde-

lati, kako bo uporabnike vključil v proces, kako jih bo spodbujal k sodelovanju in nazadnje tudi v kolikšni meri jih bo upošteval.

Razvojni cikel je zato dobro že v samem začetku pripraviti tako, da ima uporabnik v njem čimvečjo vlogo. Takemu pristopu pravimo *uporabniško usmerjen razvoj* (*User-centered design* ali *User-driven development* oz. *UDD*) [8]. Njegovo osnovno načelo je razvoj take rešitve, kakršno uporabniki želijo uporabljati, in ne take, ki bi od uporabnikov zahtevala, da svoj trenutni delovni proces spremenijo in prilagodijo rešitvi.

Uporabniško usmerjen iterativno inkrementalni razvoj je torej odlična izbira pri razvoju recepcijske programske opreme za manjše objekte. Prvi korak pri zasnovi lastne rešitve je predstavljalo odločanje, kako tak razvojni proces udejaniti. Dober in enostaven način, ki se je v svetu brezplačne in odprtokodne programske opreme pokazal za zelo učinkovitega, je postavitev spletne strani rešitve z uporabniškim forumom [9], na katerem lahko kdorkoli (tako registriran kot neregistriran uporabnik) postavi vprašanje, prijavi napako ali predlaga nadgradnjo.

V samem začetku je število prispevkov, objavljenih v forumu, majhno. Uporabnik, ki se opogumi, objavi svoje vprašanje, napako ali idejo. Če razvijalec uporabnikov prispevek v naslednji iteraciji upošteva in to tudi jasno pokaže (na primer z odgovorom v forumu ali z novico), uporabnik občuti, da je tudi sam del razvojnega procesa. Ta občutek povzroči dve pomembni posledici:

- rešitev postane uporabniku bolj pri srcu, saj se tudi sam počuti odgovornega zanjo, ter
- uporabnik dobi pogum in motivacijo za nadaljnjo uporabo foruma.

V najuspešnejših primerih lahko postane prispevek določenega uporabnika tako velik, da prične uporabnik še sam odgovarjati na vprašanja drugih uporabnikov in voditi razpravo na forumu.

4.1.2 Trislojna arhitektura

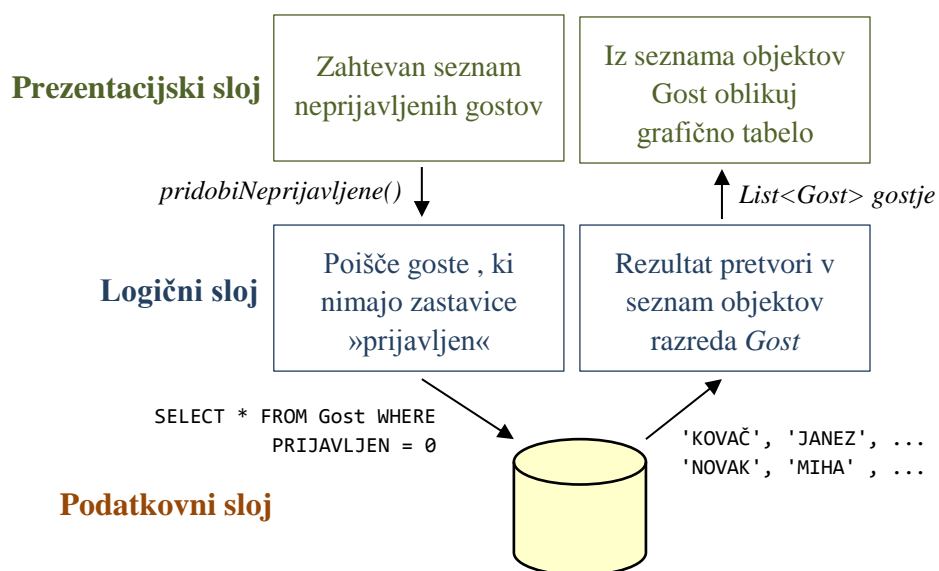
Z vidika razvijalca so glavne naloge recepcijske programske opreme hranjenje, obdelava in prezentacija podatkov. Dostop do podatkov in njihova obdelava morata biti hitra, zato je smiselno podatke hraniti na način, ki to omogoča. V recepcijskem delu so podatki občutljive narave, zato mora aplikacija pri delu z njimi ohranjati integriteto. Onemogočene morajo biti operacije, ki lahko integriteto ogrozijo, prav tako pa je potrebno pri vnosu novih podatkov paziti, da so ti skladni z načrtovano obliko.

V takem primeru je dobra praksa neposredno upravljanje s podatki omogočiti le enemu delu programske opreme (na primer razredu, knjižici ali samostojni storitvi), ostalim delom programske opreme pa upravljanje dovoliti le posredno preko njega. Takemu modelu pravimo *model odjemalec-strežnik*. Model lahko implementiramo na več načinov. Rešitev lahko na primer razdelimo na več slojev, pri čemer vsak izmed njih ponuja zbirko funkcij, s katerimi so sloji povezani med seboj. Sloj, ki se povezuje z drugim, to počne izključno preko njegovega vmesnika - o samem izvajanju v ozadju pa mu ni potrebno vedeti ničesar. Taki arhitekturi programske opreme pravimo *večslojna arhitektura* (*multitier architecture*). Če svojo rešitev razdelimo na vsaj tri sloje, zagotovimo:

- večjo razširljivost,
- večjo zmogljivost in
- večjo učinkovitost.

Negativna posledica izbire take arhitekture je večja kompleksnost programske kode, ki zahteva več napora in časa za pisanje. V primerjavi s prednostmi, ki jih arhitektura prinaša, je ta zanemarljiva, in je izbira trislojne arhitekture dobra odločitev.

Sloji imajo načeloma lahko poljubno nalogo, vendar jih običajno opredelimo kot *prezentacijski*, *logični* in *podatkovni sloj* [10]. Na sliki 4.2 so s praktičnim primerom prikazane vloge vsakega izmed njih:



Slika 4.2: Primer poteka operacije pri trислоjni arhitekturi

Prezentacijski sloj je vmesni člen med uporabnikom in logičnim slojem. Pogosto je implementiran kot grafični uporabniški vmesnik. V primeru s slike 4.2 uporabnik zahteva tabelo vseh neprijavljenih gostov. Prezentacijski sloj pokliče funkcijo *pridobiNeprijavljene()*, ki jo nudi logični sloj. Logični sloj pripravi SQL poizvedbo, ki jo posreduje podatkovnemu sloju. Upravljalni sistem podatkovne baze izvede poizvedbo in neobdelan rezultat vrne na višji sloj. Ta nato iz rezultata razbere podatke, ki jih potrebuje, in iz njih sestavi zbirko objektov razreda *Gost*, ki se vrne prezentacijskemu sloju, ki končno zbirko tudi grafično uprizori na zaslonu v obliki tabele.

4.1.3 Manj modulov, več funkcij

Poglavje 2 osnovno recepcijsko programsko opremo deli na 4 osnovne module in dodatna orodja. Najpomembnejša med njimi sta elektronska knjiga gostov in urejevalnik rezervacij. Upravljalce kanalov in blagajna sta vsekakor prav tako zelo pomembna, vendar sta bolj pogrešljiva.

V besedilu je bilo že večkrat poudarjeno, da razvijalci recepcijske programske opreme svoje rešitve z vidika modulov prepogosto razvijajo kvantitativno namesto kvalitativno. Večje število implementiranih modulov naredi rešitev privlačnejšo na prvi pogled, a če so ti funkcionalno pomanjkljivi ali slabo implementirani, je taka rešitev za uporabnika nekoristna.

Pri izbiri obsega funkcionalnosti in modulov v lastni rešitvi je bilo odločeno naslednje:

- v prvi iteraciji implementirati samo najnujnejša modula - knjigo gostov in urejevalnik rezervacij,
- izbrana modula implementirati dobro in delovanje temeljito testirati ter
- izbrana modula opremiti s funkcionalnostmi, ki jih ciljni uporabniki pogrešajo pri ostalih rešitvah.

4.2 Programski paket eGost

Bralec je v besedilu programski paket eGost že nekoliko spoznal, vendar dejstvo, da predstavlja paket edini vmesni člen med elektronsko knjigo gostov in policijo, zahteva, da razvijalec njegovo delovanje razume, preuči zahteve za pravilno uporabo in ga nazadnje zna v svojo rešitev pravilno integrirati.

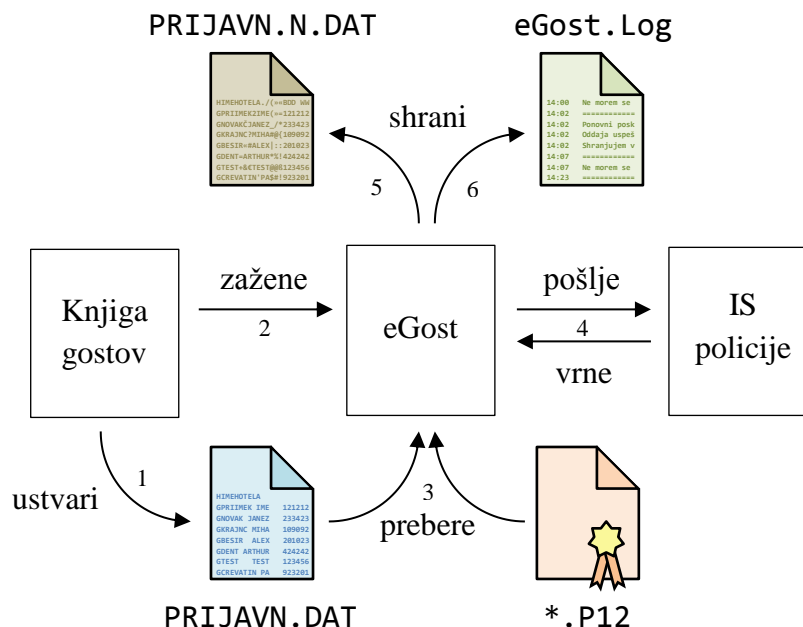
Testno verzijo paketa eGost lahko razvijalec pridobi s pisno prošnjo, ki jo pošlje policiji. Poleg programskega paketa prejme tudi nekaj informacij o njegovem delovanju ter podrobna navodila o zahtevah, ki so zakonsko predpisane.

Paket eGost je v osnovi zelo preprost. Sestavljajo ga:

- izvršljivi program `eGost.exe`,
- digitalno potrdilo `*.P12`,
- konfiguracijska datoteka `eGost.ini`,
- dnevnik dogodkov v obliki tekstovne datoteke `eGost.Log`,
- izvršljivi program `PreveriPodpis.exe` in
- navodila za uporabo `eGost.chm` in `eGost.pdf`.

Poleg naštetih datotek programski paket vsebuje še 4 direktorije:

- *Arhiv*, v katerega se shranjujejo odposlane prijavnice,
- *Oddaja*, v katerega mora elektronska knjiga gostov shraniti prijavnico za oddajo na policijo,
- *Sprejem*, v katerega se prenesejo vrnjene prijavnice pred premikom v arhiv, ter
- *uninst*, ki vsebuje program za odstranitev paketa.



Slika 4.3: Diagram postopka pošiljanja prijav gostov na policijo s paketom eGost

4.2.1 Delovanje in integracija

Program deluje na principu pošiljanja datotek na informacijski sistem policije. Postopek je grafično prikazan na sliki 4.3, koraki, ki so označeni s številkami, so pa naslednji:

1. Elektronska knjiga gostov mora iz podatkov gostov, ki jih želi uporabnik prijaviti, tvoriti besedilno datoteko predpisane oblike, jo poimenovati **PRIJAVN.DAT** ter shraniti v direktorij *Oddaja*;
2. V naslednjem koraku mora uporabnik pognati izvršljivo datoteko paketa eGost (kot kaže slika 4.3, lahko zagon sproži tudi knjiga gostov), vpisati geslo, ki ga je pridobil pri policiji, in oddajo potrditi;
3. eGost preveri, ali se v direktoriju *Oddaja* nahaja kakšna datoteka. Če je direktorij prazen, prikaže obvestilo, da ni datotek za oddajo, in se

po potrditvi ustavi;

4. Vse datoteke, ki jih eGost najde, digitalno podpiše in pošlje na strežnik, ki jih tudi sam digitalno podpiše in vrne;
5. Uporabniku se prikaže potrdilo o oddaji z vsebino oddane datoteke, ki ga lahko natisne, podpisane datoteke pa eGost premakne v direktorij *Arhiv*;
6. V tekstovno datoteko **eGost.Log** se ob tem zapisujejo informacije o prenosu, med drugim tudi, ali je bila oddaja uspešna ali ne.

Kje lahko pride do težav?

V četrtem koraku opisanega postopka paket eGost ne preverja vsebine ali ustreznosti datoteke **PRIJAVN.DAT**. Če elektronska knjiga gostov ustvari popolnoma neustrezno datoteko, je ta vseeno odposlana. Neustrezna datoteka sproži napako šele v informacijskem sistemu policije po zaključeni oddaji. Uporabnik tako ne ve, da je bila odposlana datoteka neveljavna.

V takih primerih nato običajno policija uporabnika pokliče, ga opozori na napako ter opomni, da je za nepravilno vodenje elektronske knjige gostov predpisana visoka kazen. Ustreznost prijavnic mora tako elektronska knjiga gostov zagotoviti sama, sicer je njena uporaba zelo tvegana (bralec se bo morda spomnil na eno izmed opisanih obstoječih rešitev).

Druga nevarnost je lastnost paketa, da pri neuspešnem prenosu (uporabnik ima na primer slabo internetno povezavo ali informacijski sistem policije ni na voljo) prijavnice iz direktorija *Oddaja* ne odstrani. To je lahko dobrodošlo, če knjiga gostov ne preverja uspeha prenosa, saj bo v tem primeru prijavnica odposlana ob naslednjem zagonu programa eGost. Vendar pa mora v tem primeru knjiga gostov ob kreiranju nove prijavnice paziti, da v direktoriju *Oddaja* ne prepíše že obstoječih.

Boljša rešitev je implementacija preverjanja uspeha prenosa in avtomatično praznjenje direktorija *Oddaja* v primeru, da ta ni bil uspešen. V tem

primeru dobi knjiga gostov povratno informacijo, da prenos ni uspel, in vpisov gostov ne označi kot odposlanih.

Kako implementirati preverjanje uspeha prenosa?

Datoteka `eGost.Log` z dnevniškimi zapisi deluje morda najprej idealna za to opravilo, vendar ima nekaj šibkosti. V datoteko se namreč zapisuje vse, kar paket eGost počne, v obliki datuma, ure in pisnega opisa, stari zapisi pa se ne brišejo. Da bi razvijalec prišel do informacije, ali je bil prenos uspešen ali ne, mora iz datoteke izluščiti zapise, ki opisujejo potek zadnjega prenosa, iz njih pa izluščiti vrstico, ki vsebuje besedo "USPEŠNO". Če take vrstice ni, lahko sklepa, da prenos ni uspel.

Opisan pristop je potencialno nevaren. Oblika zapisov v datoteki z dnevniškimi zapisi ni predstavljena v nobeni dokumentaciji in razvijalec ne pozna vseh možnih vrst zapisov (morda obstaja nepogost scenarij, pri katerem se v datoteko zapišeta besedi "NI USPEŠNO", česar razvijalec ne more vedeti, in se posledično poruši logika preverjanja).

Alternativni pristop, ki je bil uporabljen pri lastni rešitvi, je pregled vsebine direktorija *Arhiv* pred zagonom paketa eGost ter ponovni pregled po koncu poskusa pošiljanja. Če se v direktoriju po poskusu pošiljanja nahaja nova datoteka, je lahko razvijalec prepričan, da je bila prijavnica uspešno oddana.

4.2.2 Predpisana oblika prijavnic

Ključna naloga elektronske knjige gostov je, da datoteko `PRIJAVN.DAT` pripravi popolnoma skladno z vsemi predpisi, ki jih razvijalec mora dobro poznati in razumeti.

Prijavnica mora biti zapisana kot besedilna datoteka v ASCII formatu s kodno tabelo 852. V prvi vrstici besedila mora biti podan *vodilni zapis prenočišča*, ki služi identifikaciji nastanitve, ki prijavnico pošilja, vsaka naslednja vrstica pa predstavlja zapis podatkov o gostih (za vsakega gosta ena vrstica).

polje	dolžina	vsebina
oznaka vrstice	1	velika črka H
naziv prenočišča	12	velike črke in števila
kraj prenočišča	15	velike črke in števila
naslov prenočišča	20	velike črke in števila
prazno	62	62 presledkov (<i>ni obvezno</i>)

Tabela 4.1: Struktura vodilnega zapisa v datoteki PRIJAVN.DAT

polje	dolžina	vsebina
oznaka vrstice	1	velika črka G
priimek gosta	33	velike črke
ime gosta	25	velike črke
datum rojstva	8	datum oblike <i>ddmmllll</i>
kraj rojstva	25	velike črke
državljanstvo	2	oznaka države po ISO 3166
datum prijave	8	datum oblike <i>ddmmllll</i>
datum objave	8	datum oblike <i>ddmmllll</i>

Tabela 4.2: Struktura zapisa podatkov gosta v datoteki PRIJAVN.DAT

Vodilni zapis mora biti sestavljen iz 110 znakov, ki predstavljajo niz polj, vsako polje pa ima natanko določeno dolžino. Če je vrednost v polju krajša od njegove dolžine, je potrebno razliko dopolniti s praznimi znaki (presledki), da se dolžine celotnega zapisa ne spremeni. Iz enakega razloga ne sme vrednost biti daljša od dolžine polja. V tabeli 4.1 so predstavljene dolžine in vrednosti vseh polj vodilnega zapisa.

Tudi zapisi podatkov gostov morajo biti dolgi 110 znakov. Dolžine in opisi polj so podani v tabeli 4.2.

Velike črke, ki jih je dovoljeno vpisovati, tvorijo vse velike črke angleške abecede ter znaki Ä, Á, É, Í, Ö, Ó, Ő, Ü, Ú, Ŭ, Č, Ć, Đ, Š in Ž.

Pri imenih in priimkih gostov, ki jih sestavlja več besed, morajo biti

besede ločene s presledkom, poševnico ali pomišljajem.

Če elektronska knjiga zagotavlja tvorbo prijavnic v obliki, ki je bila opisana, pa še ni mogoče trditi, da je prijavnica ustrezna. Prijavnica namreč lahko vsebuje napake na vsebinski ravni (na primer če je datum rojstva večji od trenutnega datuma ali oznaka države ni skladna s standardom ISO 3166).

Kratka navodila, ki jih razvijalec prejme ob dodelitvi testne verzije programa eGost, navajajo le nekaj možnih vsebinskih napak, kot sta recimo zapis dveh presledkov med besedama, ki tvorita priimek gosta, in vpis datuma rojstva, ki je večji od trenutnega datuma. Možnosti za vsebinske napake, ki jih navodila ne navajajo, je veliko, zato mora razvijalec pri implementaciji preprečevanja napak o njih zelo dobro premisliti.

4.3 Implementacija

Lastna rešitev je implementirana kot brezplačni program z imenom Open-gost, ki ga lahko uporabniki brez registracije namestijo na svojo napravo in nemudoma pričnejo uporabljati.

V prvi različici implementira le knjigo gostov in urejevalnik rezervacij, vendar je jedro programa strukturirano tako, da ga je mogoče hitro in enostavno obogatiti z dodatnimi funkcionalnostmi.

4.3.1 Platforma in orodja

Ponudba brezplačne in/ali odprtokodne programske opreme, z uporabo katere lahko razvoj lastne rešitve bistveno pohitrimo in olajšamo, je velika. Pri razvoju rešitve, ki bo tudi sama ponujena brezplačno, je uporaba take programske opreme še toliko bolj smiselna.

Ker programski paket eGost deluje le na operacijskem sistemu Microsoft Windows, integracija s paketom pa je nujna za implementacijo elektronske knjige gostov, lahko razvijalec bodisi:

- svojo rešitev tudi sam omeji na operacijski sistem Microsoft Windows

ali

- svojo rešitev implementira tudi za druge platforme, vendar od uporabnika zahteva, da prijavnice za paket eGost iz programa izvaža in uvaža na drug računalnik, na katerem sta nameščena Microsoft Windows in paket eGost.

Ker mora imeti končni uporabnik, ne glede na izbiro prve ali druge možnosti, vsaj na enem računalniku nameščen operacijski sistem Microsoft Windows, za izbiro druge možnosti ni posebne motivacije. Poleg tega pa je operacijski sistem Microsoft Windows med uporabniki razširjen in priljubljen.

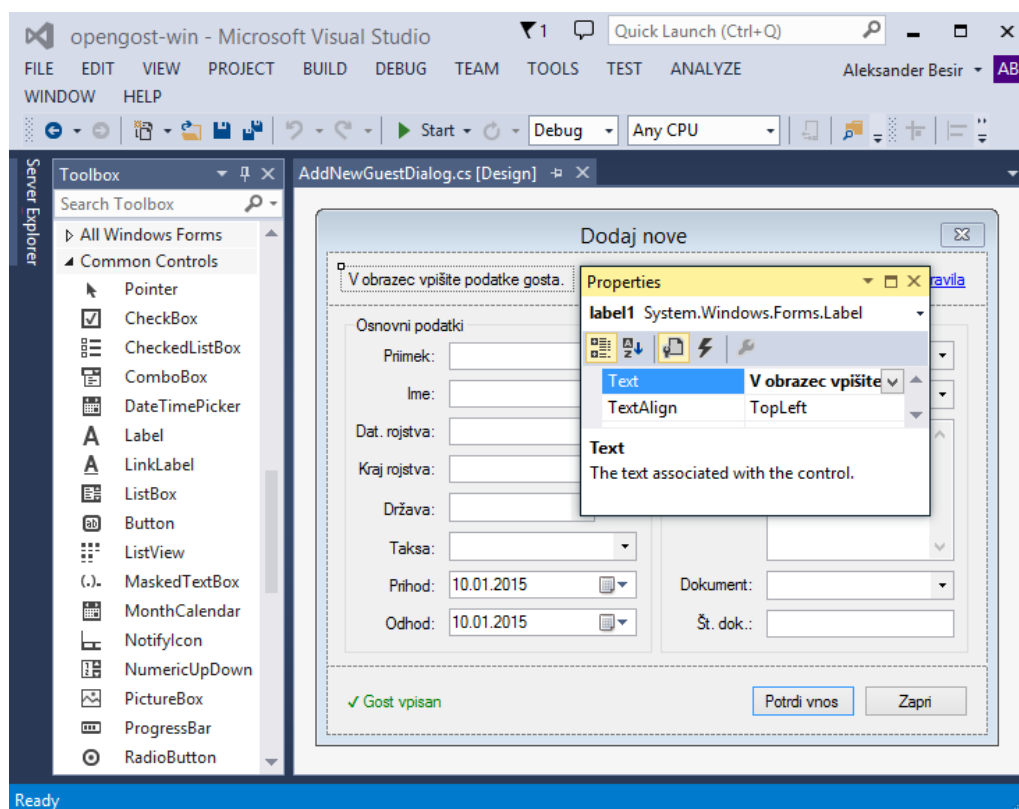
Za ciljno platformo lastne rešitve je bil tako izbran izključno operacijski sistem Microsoft Windows, in sicer njegove novejšje različice (Vista SP2, 7 SP1, 8 ter 8.1). Ta izbira je posledično omogočila, da se celoten projekt umesti v brezplačno (in delno odprtokodno) ogrodje Microsoft .NET 4.5.

Ogrodje .NET je dobra izbira za hiter in učinkovit razvoj programske opreme, saj omogoča programiranje v enem izmed več podprtih jezikov, vsebuje zelo bogato knjižico že implementiranih funkcionalnosti ter je z uporabo zmogljivega brezplačnega orodja Microsoft Visual Studio Community mogoče načrtovati in izdelovati dobre uporabniške vmesnike [11] (tudi na grafičen način, kot je vidno na sliki 4.4).

4.3.2 Podatkovni sloj

Podatkovni sloj implementira lokalna podatkovna baza v prenosljivi datoteki Microsoft Transact-SQL MDF. Sam program tako ne vsebuje sistema za upravljanje s podatkovnimi bazami (*Database Management System* oz. *DBMS*), pač pa le datoteko, ki vsebuje podatke podatkovne baze.

Namestitveni program najprej preveri, ali ima uporabnik na svojem računalniku že nameščen brezplačni sistem za upravljanje s podatkovnimi bazami SQL Server 2012 Express LocalDB. Če je sistem že nameščen, lahko program Opengost ob zagonu lastno podatkovno bazo preprosto pripne sistemu za



Slika 4.4: Orodje Microsoft Visual Studio Community omogoča grafično izdelavo uporabniškega vmesnika

upravljanje in z njo dela. Če sistem ni zaznan, ga namestitveni program namesti.

Ena izmed prednosti take implementacije je, da rešitvi ni potrebno nameščati novega sistema za upravljanje s podatkovnimi bazami, če to ni potrebno. Veliko obstoječih aplikacij v svojem ozadju že uporablja SQL Server 2012 Express LocalDB, zato obstaja možnost, da je sistem na uporabnikovem računalniku že nameščen. Druga prednost je v tem, da se vse spremembe nad podatki sproti zapisujejo v datoteko MDF. Podatkovna baza je s tem izredno prenosljiva, saj jo je mogoče kopirati na drugo napravo s preprostim kopiranjem datoteke MDF¹. Možnost takega kopiranja pa hkrati tudi zelo poenostavi proces izdelovanja varnostnih kopij.

Posodabljanje

Opengost ob prvi namestitvi kreira podatkovno bazo na uporabnikovi napravi. Ta vsebuje implementacijo podatkovnega modela v obliki praznih tabel, pogledov in ostalih SQL objektov. Pri posodobitvi programa se pričakuje, da tabele v podatkovni bazi niso več prazne, zato se posodobita le logični in prezentacijski sloj.

Ker je razvoj programa iterativno inkrementalen, se je potrebno zavedati, da se bo podatkovni model v prihodnjih iteracijah spreminjal (redko, a zagotovo), kar zahteva dodajanje novih tabel v podatkovno bazo in spreminjanje strukture obstoječih.

Pri novih namestitvah to ne predstavlja težave, saj program ob namestitvi kreira podatkovno bazo, v kateri je najsodobnejša različica podatkovnega modela že implementirana. To seveda ne velja pri posodobitvah, saj polne

¹Poznavalca bi lahko zapisana trditev zmotila, saj lokalno podatkovno bazo za SQL Server 2012 Express LocalDB v resnici tvorita dve datoteki (poleg datoteke MDF tudi datoteka LDF). Datoteka LDF služi zapisovanju dnevnika transakcij in se lahko uporabi v primeru, če želimo stanje podatkovne baze povrniti na natanko določen trenutek v preteklosti. Če aplikacija te funkcionalnosti ne potrebuje, lahko datoteko LDF ignorira. Ob delu s podatkovno bazo, ki nima pripadajoče LDF datoteke, SQL Server 2012 Express LocalDB kreira novo (prazno) datoteko LDF samodejno.

podatkovne baze ne smemo preprosto nadomestiti s prazno. V tem primeru je potrebno podatkovni model posodobiti posredno preko logičnega sloja. Na voljo sta 2 možnosti:

- podatkovni sloj ob posodobitvi bodisi kreira novo podatkovno bazo z novim podatkovnim modelom in vanj prenese vse podatke iz tabel stare podatkovne baze ali
- obstoječo podatkovno bazo uredi z SQL ukazi za manipulacijo strukture tabel.

Oba pristopa imata prednosti in slabosti, pri čemer sta glavni slabosti prvega, da je proces lahko pri velikem številu podatkov dolgotrajen in da mora podatkovni sloj znati podatke prebrati iz vseh različic preteklih podatkovnih modelov. Slabost drugega pristopa je v tem, da če razvijalec poskusi obstoječo tabelo spremeniti na način, ki lahko ogrozi integriteto podatkov v njej, sistem za upravljanje z bazo operacije ne dovoli, kar posledično onemogoči posodobitev.

Prvi pristop od razvijalca zahteva veliko programiranja, medtem ko drugi nekaj več testiranja. Vendar pa le drugi pristop zagotavlja, da pri posodabljanju ne more priti do izgube ali okvare obstoječih podatkov, kar je pri delu z občutljivimi podatki nujno, in je zato bil tudi izbran.

4.3.3 Logični sloj

Logični sloj predstavlja jedro programa Opengost in implementira:

- vse funkcije, ki so potrebne za delo s podatki podatkovnega sloja, in
- ontologijo [12] - vse razrede, ki te podatke abstrahirajo v človeku bolj naravne enote, razmerja med njimi ter pravila.

Razni podatki o gostu se tako lahko na podatkovnem sloju fizično nahajajo v več različnih tabelah, vendar jih logični sloj zna poiskati, obdelati in

združiti ter iz njih kreirati nov objekt tipa *Gost*. Do gostovega imena je nato mogoče priti preko razredne lastnosti *Ime* s preprostim klicem *gost.Ime*.

Razredi, s katerimi so podatki predstavljeni, vsebujejo funkcije za izvajanje poizvedovanja po podatkih in operacij nad njimi. Nekatere funkcije, kot so branje, shranjevanje sprememb, brisanje in kreiranje novih instanc (poznane tudi kot operacije *CRUD* [13]), so skupne vsem razredom in implementirane v generičnem razredu, ki ga ostali dedujejo. Druge so smiselne samo pri določenih razredih in so implementirane na lokalni ravni.

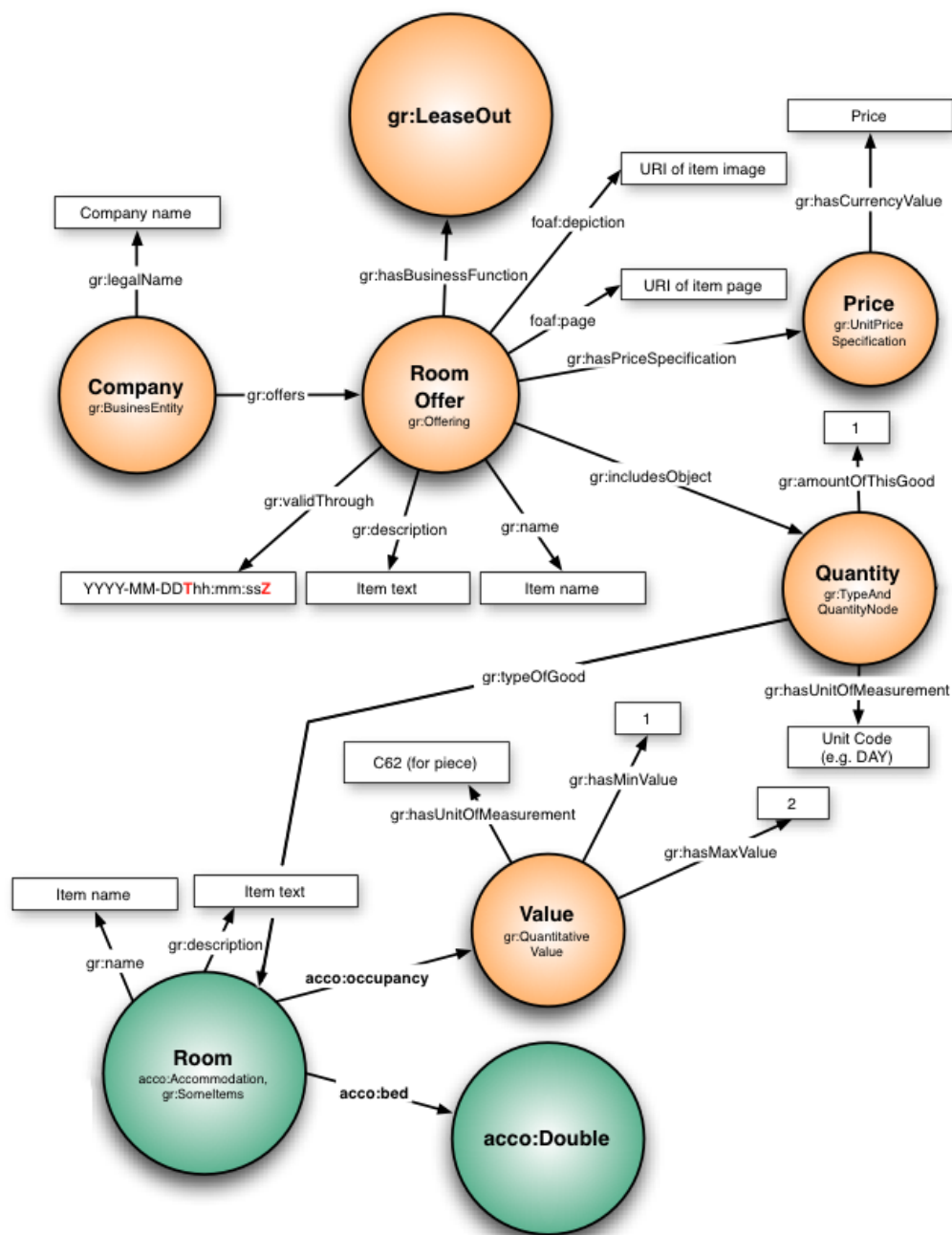
Implementacija teh razredov in funkcij zahteva dobro poznavanje recepcijskega in hotelskega dela na splošno. Razvijalec, ki želi imeti dobro osnovo za nadaljnji razvoj, mora veliko časa in premisleka posvetiti razvoju ontologije problemske domene. K sreči je s popularizacijo razmeroma nove ideje semantičnega spleta [14] veliko raziskovalcev pričelo poskušati razviti dobre ontologije za najrazličnejša področja. Eden takih je tudi Martin Hepp, ki je leta 2013 na innsbruški univerzi razvil dovršeno ontologijo za področje nastanitev [15]. Slika 4.5 vsebuje le del te ontologije v obliki primera uporabe.

Pri implementaciji logičnega sloja tako Heppova ontologija bistveno olajša in pohitri razvoj. Čeprav je veliko širša od tiste, ki jo potrebujeta knjiga gostov in urejevalnik rezervacij, nudi odličen vpogled v domeno in omogoča pripravo lastne ontologije tako, da jo je v prihodnosti mogoče razširiti brez prevelikega napora.

4.3.4 Prezentacijski sloj

Motivacija za velik pomen dobro implementiranega prezentacijskega sloja je bila že podana v podpoglavju 4.1. Donald Norman v svoji odlični knjigi *The Design of Everyday Things*, ki ni omejena samo na uporabniške vmesnike računalniških programov in še danes velja za eno najpomembnejših knjig s področja dizajna, v ospredje postavlja filozofijo, da mora uporabniški vmesnik [16]:

- v čimvečji meri posnemati resnično delo v resničnem svetu,



Slika 4.5: Primer uporabe Heppove ontologije za dvoposteljno sobo z določeno ceno za eno ali dve osebi

- biti tak, kakršnega želijo in pričakujejo uporabniki, ter
- vsebovati le tisto, kar je uporabnikom v korist, vse ostalo (pa četudi estetičnega namena) odstraniti.

Uporabnik ne sme biti podrejen uporabniškemu vmesniku, vendar ravno obratno. Če mora uporabnik za realizacijo kakega opravila pripadajočo funkcijo iskati, je uporabniški vmesnik slab. Kontrole na grafičnem vmesniku (gumbi, vnosna polja in podobno) morajo delovati in izgledati tako, kot uporabnik po izkušnjah iz uporabe drugih vmesnikov pričakuje.

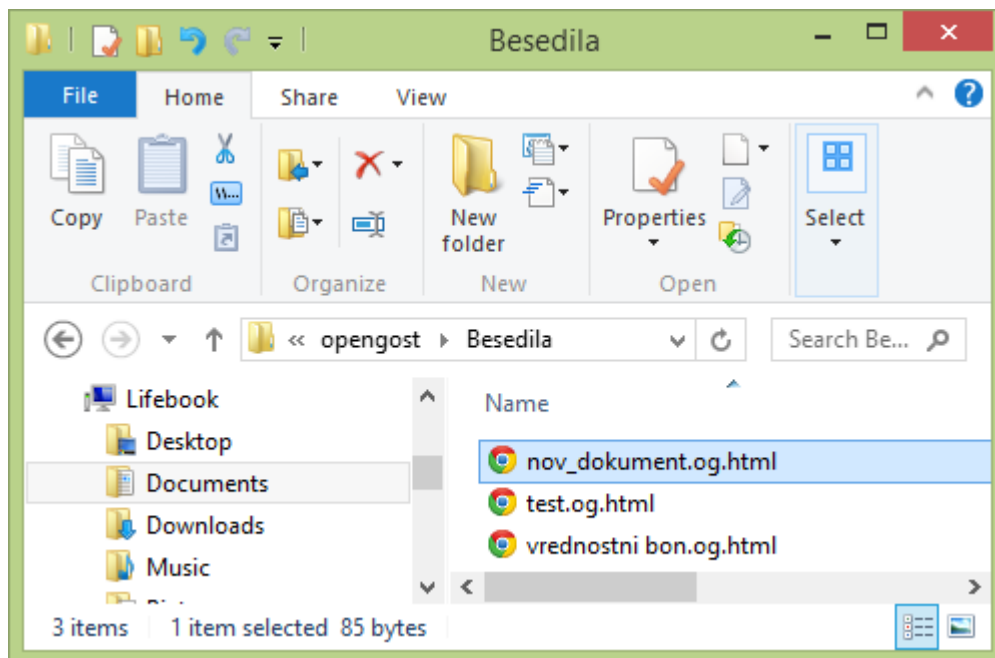
Pri implementaciji uporabniškega vmesnika lastne rešitve so bila ta načela upoštevana. Seveda jih v določenih primerih ni mogoče popolnoma zadovoljiti, vendar se lahko razvijalec potruди, da jih zadovolji, kolikor je le mogoče. S tem namenom je potrebno preučiti, kaj velja za povprečnega uporabnika recepcijske programske opreme za manjše nastanitvene objekte.

Povprečen uporabnik ciljne skupine:

- dobro pozna grafični uporabniški vmesnik operacijskega sistema Microsoft Windows, saj je le-ta na trgu najbolj razširjen,
- pozna grafična uporabniška vmesnika programov Microsoft Office Word in Excel (ali podobnih pisarniških programov),
- je navajen razne obrazce hitro izpolnjevati s premikanjem med polji s pomočjo tipke *TAB* in
- zna s spletne strani prenesti namestitveni program in ga pognati.

Za zgled je bil tako izbran sam uporabniški vmesnik operacijskega sistema Windows, čigar primer okna prikazuje slika 4.6. S slike lahko razberemo, da okno sestavljajo:

- glavni menu (na vrhu), ki določa vsebino podmenuja,
- podmenu, ki širinsko zavzame ves prostor, ki je na voljo, in vsebuje male in velike gumbe, ki jih po potrebi združi v skupine,

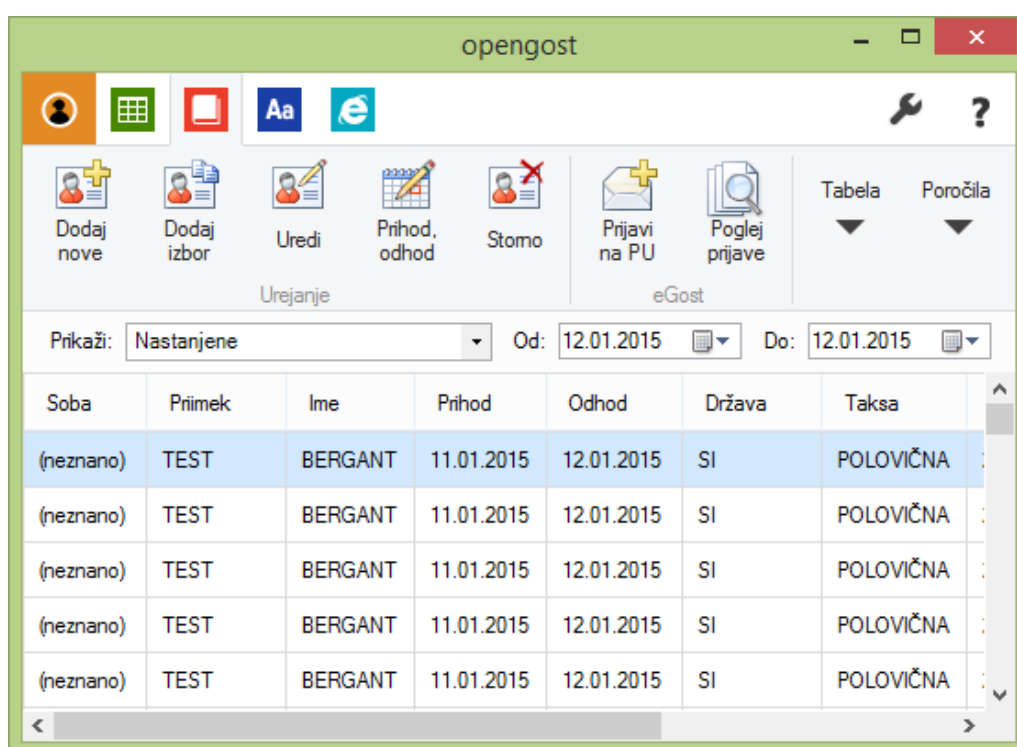


Slika 4.6: Primer okna v grafičnem vmesniku operacijskega sistema Windows 8.1

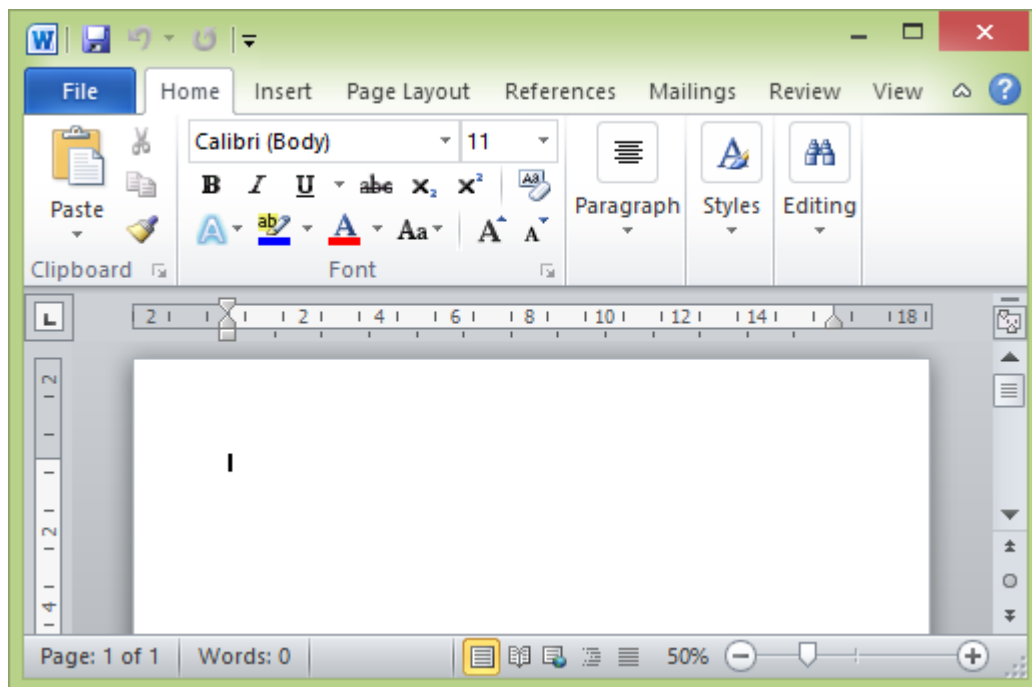
- vrstica z orodji za navigacijo in iskanje ter
- vsebina, ki grafično predstavlja nabor izbranih podatkov in zavzema ves preostali prostor.

S sliko 4.7 se lahko bralec na lastne oči prepriča o podobnosti med opisanim oknom sistema Windows in oknom uporabniškega vmesnika programa Opengost. Komponente, ki jih vsebujeta, so pozicionirane enako, podmenujeta pa se širini okna prilagajata na enak način. Povprečni porabnik, za katerega smo predpostavili, da okolje Windows že dobro pozna, tako intuitivno ve, kako z vmesnikom rokovati in kaj od njega lahko pričakuje.

Enak slog uporabljenih ikon gumbov, na katere je uporabnik navajen tudi iz programov Word in Excel, povzroči občutek domačnosti in uporabnik lahko njihovo funkcijo predvidi že brez branja napisov pod njimi.



Slika 4.7: Primer okna grafičnega uporabniškega vmesnika Opengost



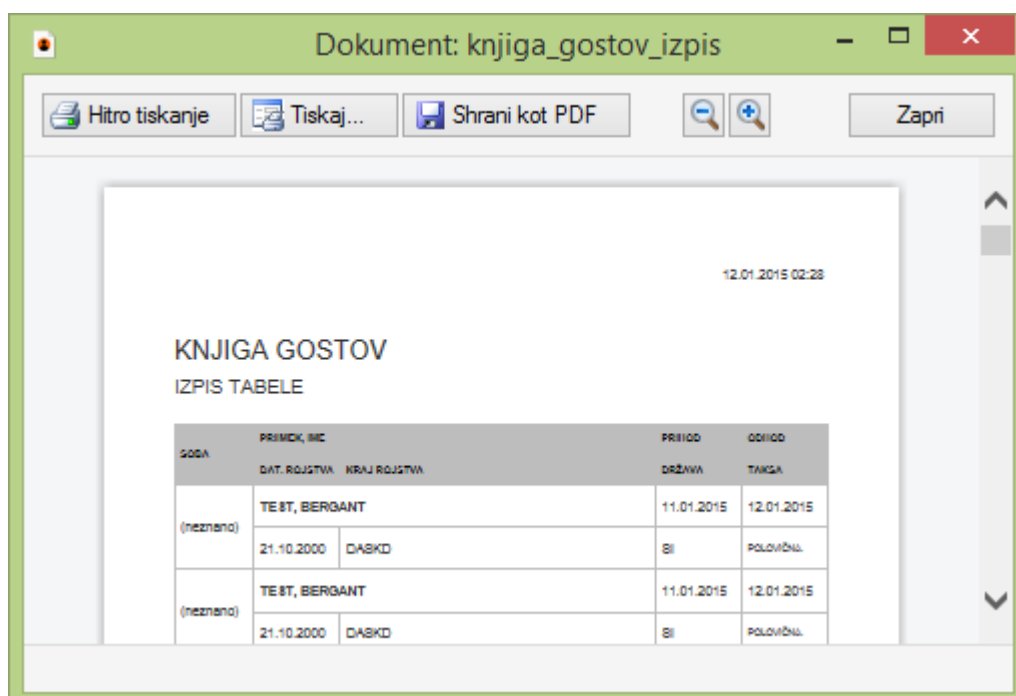
Slika 4.8: Uporabniški vmesnik programa Microsoft Office Word

Uporabniški vmesnik programa Word, ki je prikazan na sliki 4.8, služi kot zgled za razvoj lastnega urejevalnika dokumentov in prikazovalnika poročil, ki ga ponazarja slika 4.9.

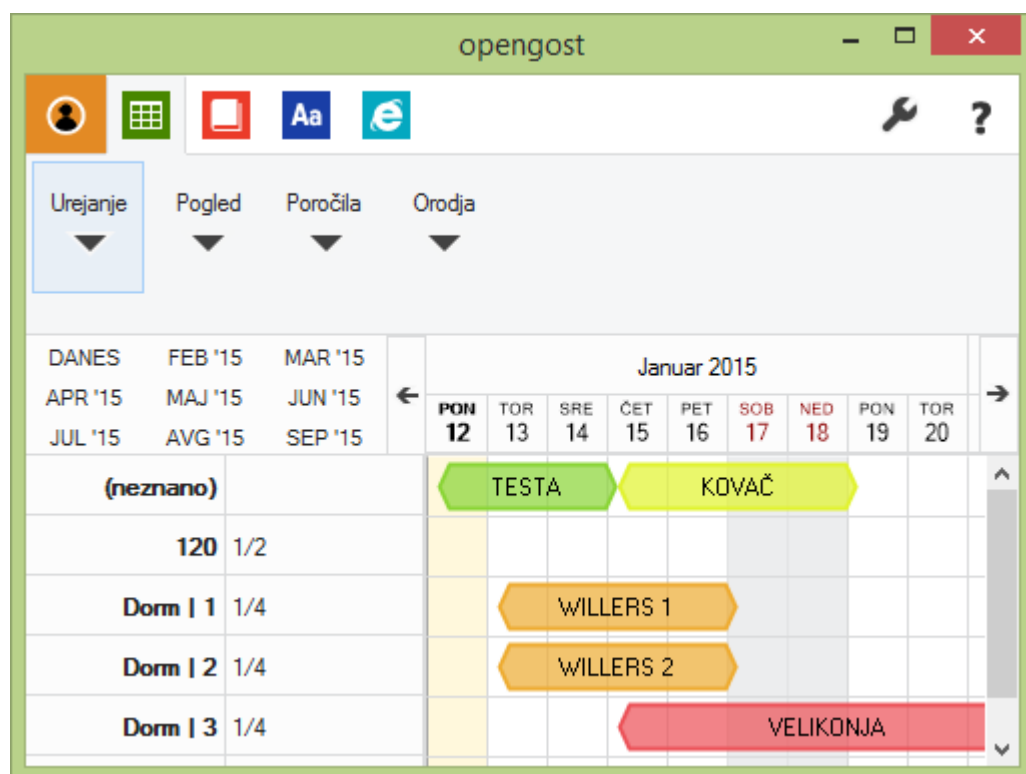
Barve in oznake, s katerimi so predstavljeni razni podatki, so mehke in nežne za oči. V nekaterih primerih - sploh v nekoliko večjih nastanitvenih objektih - uporabnik z recepcijsko programsko opremo dela neprestano, zato so preagresivne barve lahko moteče. Primer je viden na sliki 4.10.

4.4 Moduli in funkcionalnosti

Pri utemeljitvi načela *manj modulov, več funkcij* v razdelku 4.1.3 je kvaliteta modulov postavljena pred njihovo kvantiteto. Čeprav sta v prvi iteraciji razvoja lastne rešitve - poleg nekaj dodatnih orodij, namestitvenega programa in spletne strani - implementirana le dva najnujnejša modula, sta izdelana



Slika 4.9: Prikazovalnik poročil v programu Opengost



Slika 4.10: Mehke in neagresivne barve v mreži urejevalnika rezervacij programa Opengost

Soba	Ime	Prihod	Odhod	Država	Taksa	Dat. rojstva	Kraj rojstva	Prijavljen
123	MOČNIK NIKA	03.01.2015	05.01.2015	NL	POLNA	31.12.2008	NEZNANO	✓
123	MOČNIK MARKO	03.01.2015	05.01.2015	NL	POLOVIČNA	14.08.1980	NEZNANO	✓
123	MOČNIK VESNA	03.01.2015	05.01.2015	NL	OPROŠČENA	10.09.1978	NEZNANO	✓
124	ČUK JANEZ	02.01.2015	08.01.2015	SM	OPROŠČENA	11.06.2001	NEZNANO	✓
124	ČUK TEREZJA	02.01.2015	08.01.2015	SM	POLOVIČNA	15.09.1961	NEZNANO	✓
124	ČUK DARKO	02.01.2015	08.01.2015	SM	POLOVIČNA	27.10.1964	NEZNANO	✓
124	ČUK TANJA	02.01.2015	08.01.2015	SM	POLOVIČNA	04.06.1965	NEZNANO	✓
125	BOGATAJ DARKO	31.12.2014	05.01.2015	TO	POLOVIČNA	26.10.2008	NEZNANO	✓
125	BOGATAJ MATIC	31.12.2014	05.01.2015	TO	POLNA	21.11.2006	NEZNANO	✓
125	BOGATAJ KLEMEN	31.12.2014	05.01.2015	TO	POLNA	04.07.1977	NEZNANO	✓
126	PRIMOŽIČ EVA	03.01.2015	05.01.2015	PA	POLOVIČNA	14.10.1981	NEZNANO	✓
127	URŠIČ UROŠ	02.01.2015	08.01.2015	PT	OPROŠČENA	20.12.1989	NEZNANO	✓
127	URŠIČ BLAŽ	02.01.2015	08.01.2015	PT	POLOVIČNA	01.07.1988	NEZNANO	✓

Slika 4.11: Uporabniški vmesnik knjige gostov

natančno, zanesljivo in ponujata funkcije, ki so namenjene reševanju tudi manj pogostih scenarijev iz resničnega recepcijskega dela.

V nadaljevanju so opisane funkcionalnosti, ki jih modula omogočata, ter dodatna orodja, namestitveni program in spletna stran za vključitev uporabnikov v proces razvoja.

4.4.1 Knjiga gostov

Knjiga gostov (na sliki 4.11) je grafično predstavljena kot tabela, ki jo uporabnik lahko pregleduje, sortira, filtrira, v njej označuje vrstice in nad njimi izvaja operacije.

Vrstice predstavljajo posamezne vpise gostov, v stolpcih pa so zajeti podatki, ki so bili vpisani. Ikona v zadnjem stolpcu nakazuje, ali je bil vpis tudi prijavljen na policijo ali ne.

Slika 4.12: Obrazec za vpis podatkov gosta

Vpisovanje novih gostov

Uporabnik sproži proces vpisovanja novih gostov s klikom na ustrezen gumb. Vpis izvede tako, da izpolni polja v obrazcu, ki je prikazan na sliki 4.12. Obrazec je prilagojen hitremu vpisovanju, saj omogoča premikanje med polji s tipko *TAB*, ob vpisu datuma rojstva samodejno predlaga višino turistične takse, ob potrditvi vpisa, ki ga je mogoče izvesti tudi s tipko *Enter*, pa obrazca ne zapre, ampak ga pripravi za vpis naslednjega gosta, pri čemer nekatera polja (kot na primer priimek in ime) počisti, druga (kot na primer država, prihod in odhod) pa ne.

Že med vpisovanjem vrednosti v posamezna polja program v ozadju spremlja, ali so vpisane vrednosti dovoljene in smiselne. Obrazec tako ne dovoli vpisovanja datumov, ki niso realni, ne dovoli vpisa datuma rojstva pred le-

tom 1900 in po trenutnem datumu, preprečuje vpisovanje dveh zaporednih presledkov med besedama, ki tvorita gostov priimek, preverja, ali je datum odhoda večji od datuma prihoda in ali so izpolnjena vsa polja, ki so nujna za prijavo gostov na policijo.

Uporabnik je lahko tako prepričan, da so podatki, ki jih program sprejme, zmeraj skladni z zakonodajo in da pri prenosu na policijo ne bodo povzročili napake. Zmoti se lahko le pri njihovi resničnosti, če podatkov z dokumenta ne pretipka pravilno.

Vpisovanje gostov, ki so že bili nastanjeni

Vsak nastanitveni objekt ima ponavadi skupino gostov, ki v objektu večkrat prenočujejo. Takim gostom pravimo *stalni gostje*, ki so za nastanitev še posebej pomembni, saj predstavljajo vir rednih nočitev, na katere lahko nastanitev računa, in se mora potruditi, da jih obdrži.

Stalni gostje se svoje pomembnosti velikokrat zavedajo, zato so v primerjavi z ostalimi gosti bolj zahtevni in od nastanitve pričakujejo, da jih ta pozna in da se jim bolj prilagaja. Gost, ki nastanitev obiskuje večkrat letno, je lahko nezadovoljen, če nastanitev od njega pri vsakem prihodu zmeraj zahteva osebni dokument.

Elektronska knjiga gostov zato mora omogočati hitro iskanje nekoč že vpisanih podatkov in njihovo ponovno uporabljivost. V programu Opengost ima uporabnik za to opravilo 2 možnosti:

- podatke individualnega gosta poišče z vpisom dela njegovega priimka v polje *Priimek* na obrazcu za vpis novega gosta ter klikne na manjši gumb ob polju ali
- podatke večjega števila gostov, ki bivajo skupaj, poišče v tabeli, označi vrstice, ki tem gostom pripadajo, in sproži postopek ponovnega vpisa.

Pri uporabi prve možnosti program v podatkovni bazi pošče vse vpise, pri katerih priimek vsebuje vpisan niz znakov, ter uporabniku rezultate prikaže v novem oknu (na sliki 4.13).

Rezultat iskanja po arhivu gostov

×

Rezultati:

Povezani gostje:

Preimek	Ime	Dr	Dat. roj.
LOGAR	NEJC	CX	03.09.2014
LOGAR	JOŽEF	CX	17.06.1970

Preimek	Ime	Dr	Dat. roj.
LOGAR	NEJC	CX	03.09.2014
IVANČIČ	MARJANA	CX	05.08.1962
LOGAR	JOŽEF	CX	17.06.1970

Izberi

Prekliči

Ponoven vpis

Gostje bodo ponovno vpisani. Določite skupni datum prihoda, odhoda in sobo. Preverite, ali je pri posameznem gostu potrebno popraviti takso.

Gost	Datum rojstva	Taksa
NEMEC ALOJZ	27.05.1967	POLNA
NEMEC ANTONIJA	22.06.1992	POLNA

Soba: 120

Datum prihoda: 01.02.2015

Datum odhoda: 02.02.2015

Potrdi Prekliči

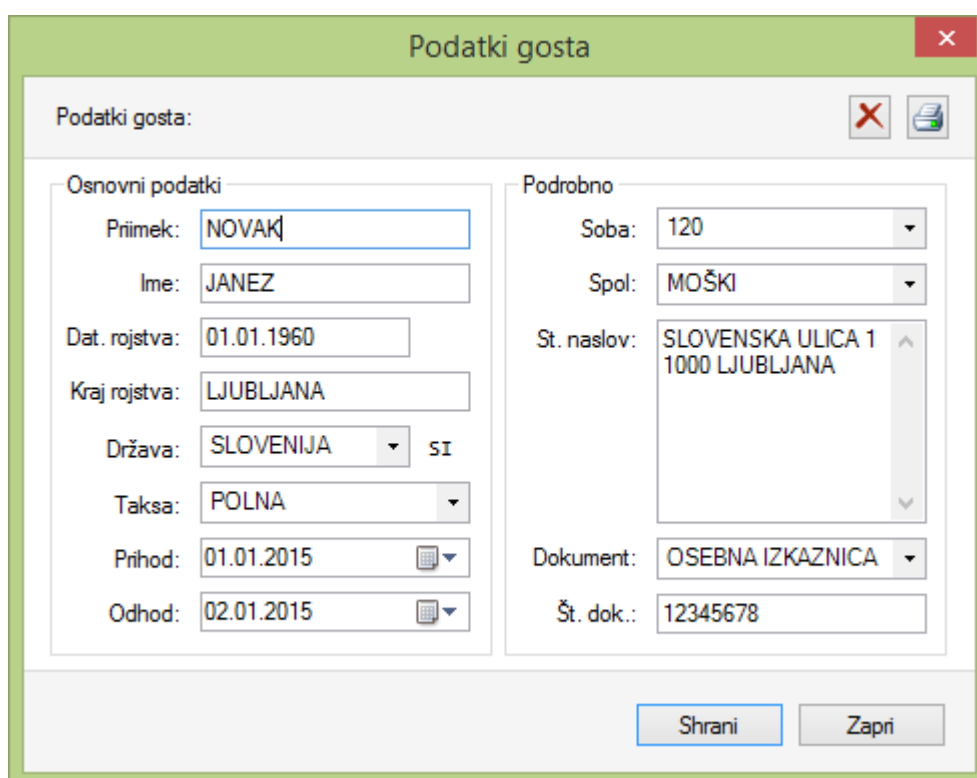
Slika 4.14: Okno za hitri ponoven vpis gostov, ki so že bivali skupaj

Urejanje posameznega vpisa

Podatke vpisanih gostov je pogosto potrebno urediti. Gost lahko na primer spremeni datum odhoda (skrajša ali podaljša svoje bivanje), uporabnik pa lahko ugotovi, da je pri vpisovanju gostovega imena pomotoma vpisal napačno črko. V vseh takih primerih je potrebno vpise popraviti in jih ponovno poslati na policijo, če so ti že bili prijavljeni.

Program tako nudi možnost urejanja že prijavljenih kot tudi neprijavljenih vpisov, kar lahko uporabnik izvede s klikom na ustrezni gumb ali dvakratnim klikom na pripadajočo vrstico v tabeli. Vpisani podatki se prikažejo v oknu z obrazcem (slika 4.15), ki je vizualno enako obrazcu za vpis novih gostov.

Uporabnik lahko podatke pregleda, uredi, natisne ali stornira, če ugotovi, da je celoten vpis neveljaven. Po potrditvi sprememb program preuči, kateri podatki so bili spremenjeni in ali je prišlo do sprememb, ki jih je potrebno



Podatki gosta

Podatki gosta:

Osnovni podatki

Preimek: NOVAK

Ime: JANEZ

Dat. rojstva: 01.01.1960

Kraj rojstva: LJUBLJANA

Država: SLOVENIJA SI

Taksa: POLNA

Prihod: 01.01.2015

Odhod: 02.01.2015

Podrobno

Soba: 120

Spol: MOŠKI

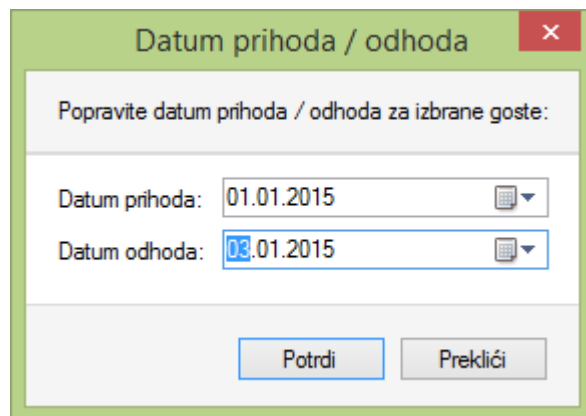
St. naslov: SLOVENSKA ULICA 1
1000 LJUBLJANA

Dokument: OSEBNA IZKAZNICA

Št. dok.: 12345678

Shrani Zapri

Slika 4.15: Obrazec za popravljanje že vpisanih podatkov



Slika 4.16: Obrazec za hitro spreminjanje datumov prihoda in odhoda za več vpisov hkrati

javiti policiji. Ob spremembi spola ponovna prijava gosta ni potrebna, saj policija tega podatka ne zahteva, ob spremembi datuma odhoda pa je. Na podlagi ugotovitev program vpis označi kot spremenjen in spremembe posreduje ob naslednji prijavi gostov na policijo.

Urejanje večjega števila gostov hkrati

Velika verjetnost je, da datuma odhoda ne spremeni le eden izmed gostov, vendar vsi gostje, ki prenočujejo skupaj. V takem primeru je popravljanje datuma odhoda za vsak vpis posebej lahko zamudno opravilo. Program zaradi tega omogoča, da uporabnik v tabeli izbere več vrstic in sproži proces skupinskega urejanja datumov prihoda in odhoda. V pripadajočem oknu (slika 4.16) lahko nastavi nove vrednosti datumov in spremembe potrdi v enem samem koraku.

Lahko se zgodi, da je med izbranimi vpisi tudi tak, ki nima enakih datumov prihoda in odhoda kot ostali. V tem primeru dobi uporabnik obvestilo, da izbrani vpisi niso datumsko usklajeni, vendar mu dovoli nadaljevati, če uporabnik tako želi.

Pošiljanje podatkov na policijo

Močna integracija s programom eGost poskrbi za to, da uporabniku ni potrebno prijavnic za policijo ročno izvoziti, premakniti v ustrezen direktorij, pognati proces pošiljanja in ugotoviti, ali so bile prijave uspešno odposlane ali ne. S klikom na ustrezen gumb program pripravi prijavnico predpisane oblike in jo shrani v ustrezen direktorij, nato pa samodejno požene postopek pošiljanja. Po končanem procesu ugotovi, ali je pošiljanje bilo uspešno, in ustrezno označi vpise gostov v tabeli kot prijavljene ali ne.

Uporabniku tako ni potrebno poznati delovanja paketa eGost - mora ga le namestiti, za pravilno delo z njim pa poskrbi Opengost sam.

Pregled odposlanih prijavnic

Pregled odposlanih prijavnic s paketom eGost je sicer mogoč tudi s paketom samim, vendar ta ne omogoča filtriranja z izbiro časovnega obdobja in vsebino prijavnic prikaže na manj pregleden tekstovni način.

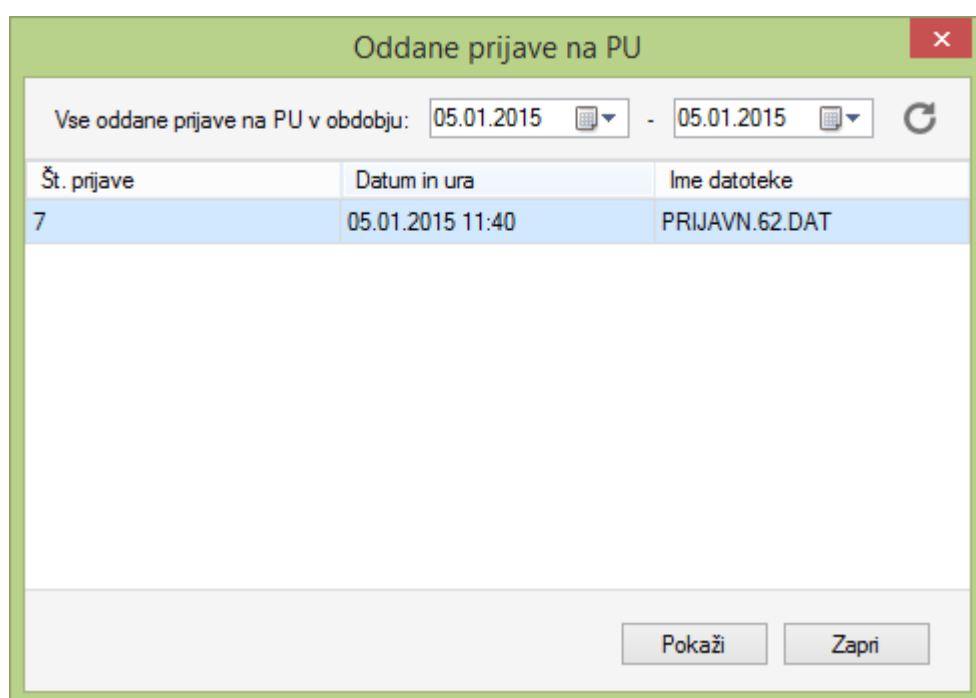
Manjkajoče funkcionalnosti so zato implementirane v lastni rešitvi. S klikom na ustrezen gumb lahko uporabnik pregleduje in išče po odposlanih prijavah (slika 4.17) in prijave tudi pregleduje v poročilu z oblikovano in urejeno tabelo.

Pregled podatkov in iskanje

Večkrat je bilo že omenjeno, da je podatke v tabeli mogoče filtrirati, sortirati in po njih iskati. To uporabniku omogoča vrstica polj pod podmenujem. Sortiranje je mogoče s klikom na ime ustreznega stolpca.

Uporabnik ima pri izbiri načina filtriranja 4 možnosti:

- prikaz vpisov vseh nastanjenih gostov v izbranem obdobju (tako tistih, ki so v izbranem obdobju prišli, kot tistih, ki so odšli, in tistih, ki so bivali),
- prikaz vpisov vseh gostov, ki so v izbranem obdobju prišli,



Slika 4.17: Seznam odposlanih prijavnic z možnostjo filtriranja, sortiranja in pregledovanja vsebine prijavnic

SOBA	PRIIMEK, IME		PRIHOD	ODHOD
	DAT. ROJSTVA	KRAJ ROJSTVA	DRŽAVA	TAKSA
123	MOČNIK, NIKA		03.01.2015	05.01.2015
	31.12.2008	NEZNANO	NL	POLNA
123	MOČNIK, MARKO		03.01.2015	05.01.2015
	14.08.1980	NEZNANO	NL	POLOVIČNA

Slika 4.18: Primer dela preglednega poročila, ki ga je možno izdelati in tiskati neposredno iz programa

- prikaz vpisov vseh gostov, ki so v izbranem obdobju odšli, ter
- prikaz vpisov gostov, ki jih je potrebno prijaviti na policijo.

Če uporabnik pri izbiri obdobja le-to določi z enim samim datumom, lahko pregleduje dnevne dogodke izbranega dneva.

Izvoz podatkov in tisk

Uporabnik mora imeti možnost nad podatki iz knjige gostov izvajati lastne operacije, ki jih morda elektronska knjiga gostov ne ponuja. Zato program implementira tudi orodje za izvoz podatkov v formatu CSV (*comma-separated values*), s čimer je njihova obdelava mogoča s programi, kakršni so Microsoft Office Excel, OpenOffice Calc ali StarOffice Calc. Če želi, lahko iz njih pripravi tudi pregledno poročilo in ga natisne neposredno iz programa (slika 4.18).

Z namenom dosega še večje učinkovitosti je podatke iz tabele mogoče s kombinacijo tipk *Control + C* preprosto kopirati v odložišče in jih nato prilepiti na poljubno mesto (na primer v Excelovi preglednici, tekstovni datoteki in podobnem).

		PRIHODI			NOČITVE			
		DOMAČI	TUJI	SKUPAJ	DOMAČI	TUJI	SKUPAJ	VREDNOST
POLNA	1,265 €	0	18	18	0	71	71	89,82 €
POLOVIČNA	0,633 €	0	23	23	0	102	102	64,57 €
OPROŠČENA	0,000 €	0	12	12	0	48	48	0,00 €
SKUPAJ		0	53	53	0	221	221	154,38 €

Slika 4.19: Del mesečnega poročila za občino, ki v tabeli navaja števila in zneske pobranih turističnih taks, prihodov in odhodov

Poročila

Za uporabnika je izdelava poročil, ki jih je zakonsko dolžan izdelovati in pošiljati pristojnim organom, ena najnujnejših funkcionalnosti. Nekatere obstoječe rešitve omogočajo izdelavo statistik in izračunov, ki jih morajo uporabniki nato prepisati v ustrezno obliko. Program lahko delo olajša tako, da so poročila, ki jih generira, že pripravljena za tisk in oddajo.

Poročila, ki so jih nastanitve dolžne izdelovati, je lahko bralec že spoznal v razdelku 2.1.4. V programu je implementirano izdelovanje vseh predpisanih poročil, pri čemer so le-ta pripravljena tako, da jih uporabniku ni treba prepisovati².

Na slikah 4.19 in 4.20 sta prikazana dela obeh obveznih poročil.

Knjiga gostov poleg obveznih poročil nudi tudi poročilo o dnevnem stanju sob, ki lahko osebju, ki je zadolženo za čiščenje enot, pomaga pri ugotavljanju

²Izjema je le poročilo za Statistični urad. Ta namreč od nastanitev zahteva, da podatke za vodenje statistike ročno s pisalom vpišejo v predpisan obrazec, ki je strojno čitljiv. Kljub temu pa lahko elektronska knjiga gostov uporabniku olajša delo, če podatke pripravi natanko v taki obliki, kakršen je obrazec TU/M. Nekatere obstoječe rešitve na primer število nočitev po državah izdelajo tako, da v tabeli prikažejo imena vseh držav, ob njih pa pripadajoče število nočitev. Obrazec TU/M ne vsebuje polj za vnos podatkov za vse države, poleg tega pa so nekatere združene v večje skupine (na primer *druge države in ozemlja Severne Amerike*), kar pomeni, da mora uporabnik v tabeli najti vrstice vseh držav, ki pripadajo skupini, in jih sešteti. V lastni rešitvi se ti izračuni izvedejo samodejno.

V. JUŽNA IN SREDNJA AMERIKA

Brazilija	076	0	0
Druge države Južne in Srednje Amerike	997	13	35

VI. SEVERNA AMERIKA





Kanada	124	0	0
ZDA (Združene države)	840	0	0
Druge države in ozemlja Severne Amerike	998	0	0

VII. VSE DRŽAVE

		ŠT. TURISTOV	ŠT. PRENOČITEV (SKUPAJ)
SKUPAJ	999	53	221

Skupno število prodanih razpoložljivih enot:	83
--	----

Slika 4.20: Del mesečnega ali letnega statističnega poročila za Statistični urad

	124	02.01. / 08.01.	
	125	31.12. / 05.01.	
	126	03.01. / 05.01.	
	127	02.01. / 08.01.	Menajva nekaterih gostov

Slika 4.21: Del poročila dnevnega stanja sob, iz katerega je razvidno, iz katerih enot gostje odhajajo, v katerih ostajajo in v katerih se spremeni število gostov

nju, ali gostje iz posamezne enote odhajajo, v njih ostajajo ali pa se v enotah le spremeni število gostov. Del omenjenega poročila je prikazan na sliki 4.21.

4.4.2 Urejevalnik rezervacij

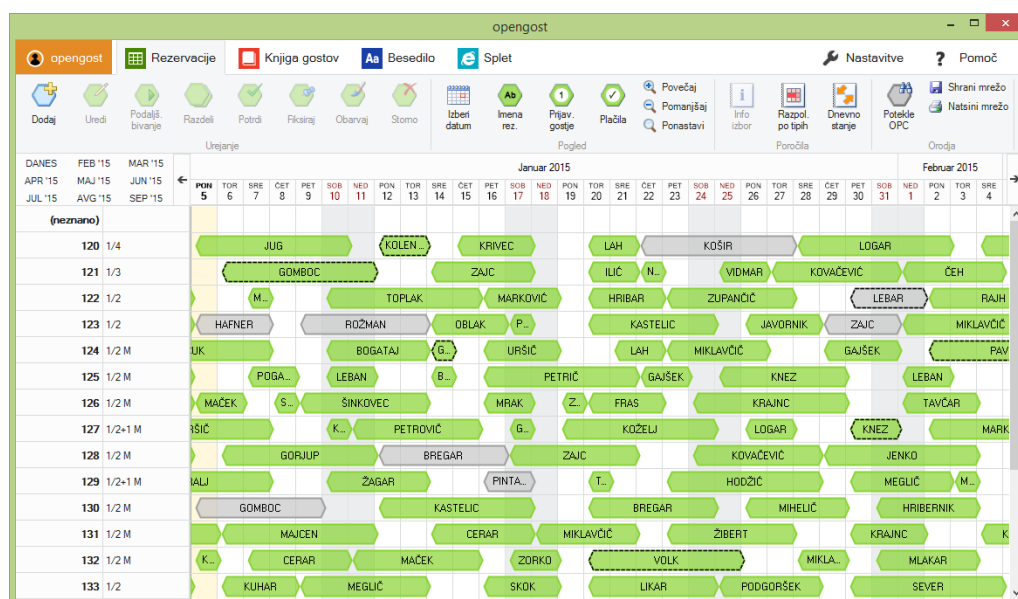
Urejevalnik rezervacij je, upoštevajoč načela dobrega grafičnega vmesnika, implementiran kot grafična mreža (slika 4.22), ki poskuša biti čimbolj podobna klasičnim fizičnim mrežam (bralec je lahko primer take že videl v drugem poglavju na sliki 2.1).

Bistvena prednost njene implementacije v programu Opengost je ta, da je zelo interaktivna. Uporabnik lahko nove rezervacije vanjo preprosto vrisuje, obstoječe pa izbira in premika kar z računalniško miško. To bistveno poenostavi proces optimiziranja zasedenosti, ki je pri nekaterih drugih implementacijah zelo zahtevno opravilo.

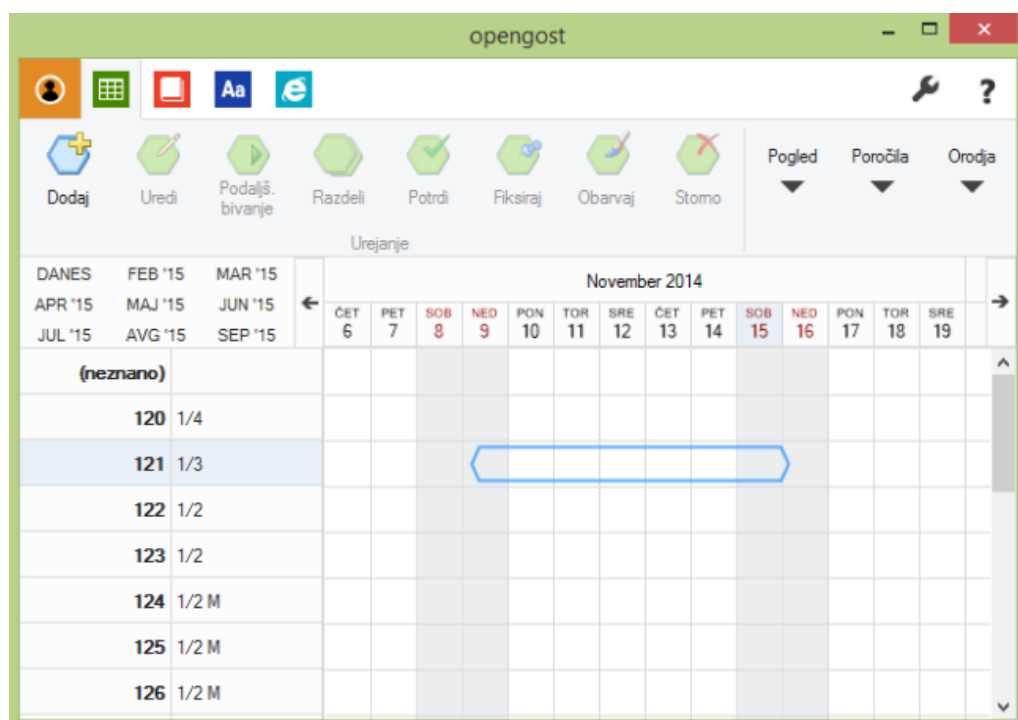
Vpis rezervacije

Proces vpisovanja novih rezervacij uporabnik prične tako, da klikne na ustrezen gumb ali novo rezervacijo vriše v mrežo (slika 4.23). Pri tem se v novem oknu prikaže obrazec za dodajanje rezervacij.

Obrazec vsebuje polja, s katerimi je mogoče rezervacijo hitro in natančno opisati:



Slika 4.22: Grafični vmesnik urejevalnika rezervacij



Slika 4.23: Uporabnik lahko novo rezervacijo vpiše z risanjem po mreži

1. ime rezervacije,
2. barvo oznake v mreži,
3. število najavljenih oseb,
4. datum prihoda in datum odhoda (urejanje teh polj samodejno posodobi prikaz števila noči, ki jih rezervacija predvideva),
5. zahteve,
6. ime dodeljene enote in ali je gost zahteval natanko njo (fiksna enota),
7. ali je rezervacija potrjena ali ne (stikalo *OPC*) in rok za potrditev,
8. ali je gost rezervacijo že plačal in do katerega dne ter
9. dodatne zabeležke.

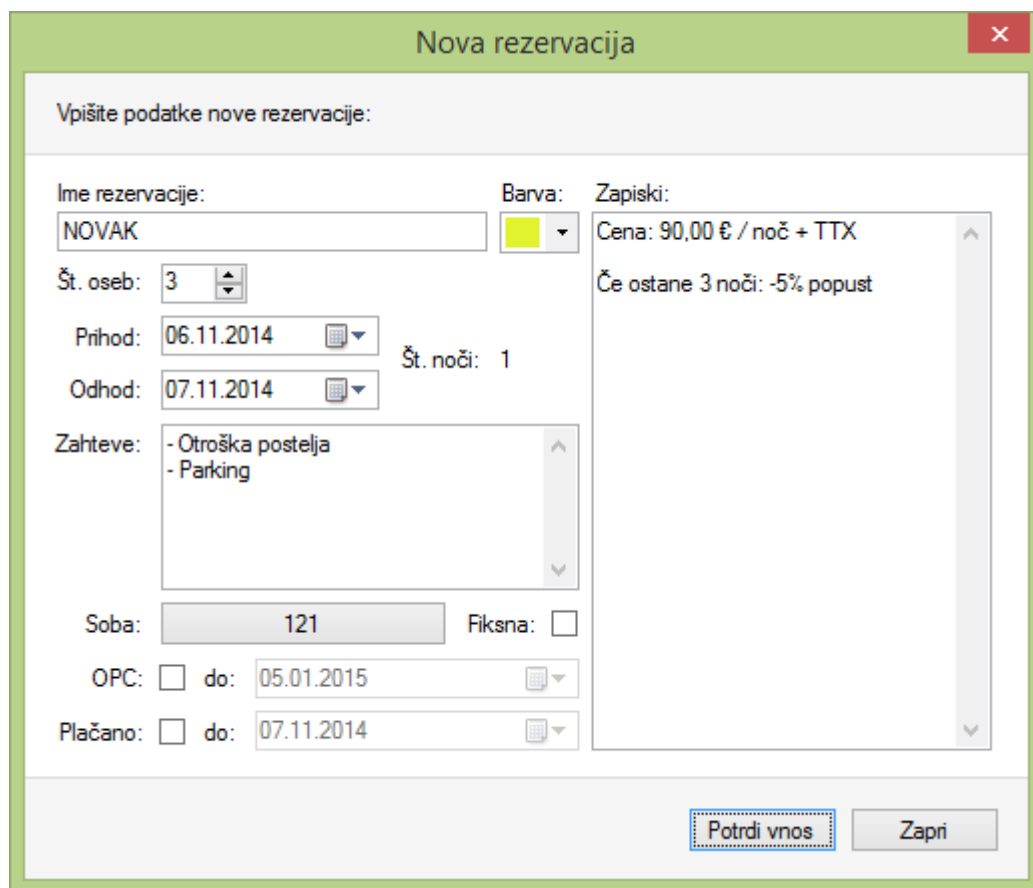
Primer izpolnjenega obrazca je prikazan na sliki 4.24.

Po potrditvi program novo rezervacijo doda v mrežo, če ta ni v konfliktu z že obstoječo rezervacijo. Obrazec se, podobno kot pri vpisovanju gostov, ne zapre, vsebina polj pa se ohrani (razen polja za izbiro sobe, ki se ponastavi). Obrazec je tako pripravljen za hiter vpis nove rezervacije na enako ime, kar je zelo uporabno, kadar želi gost rezervirati večje število sob, še posebej pa pri sprejemanju skupinskih rezervacij.

Lahko se zgodi, da poskusi uporabnik vpisati novo rezervacijo, ki se prekriva z eno ali več obstoječih. Take rezervacije program smatra kot *konflikte* in uporabniku vpisa ne dovoli, dokler jih ne razreši. Seznam konfliktov se prikaže v novem oknu (slika 4.25), kar uporabniku omogoča, da preuredi dodeljene sobe tako, da se konflikti razrešijo.

Izbira sobe s seznama prostih sob

Obrazec za dodajanje nove rezervacije vključuje pametno polje za izbiro dodeljene enote (sobe). Uporabnik tako oznake sobe ne vpiše, pač pa jo ob



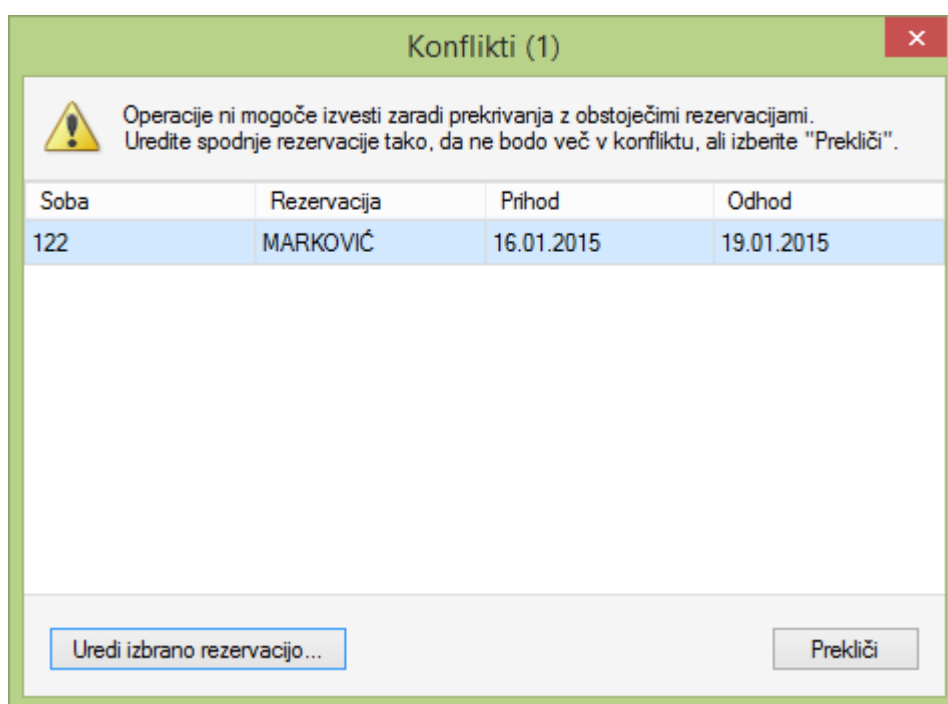
Nova rezervacija

Vpišite podatke nove rezervacije:

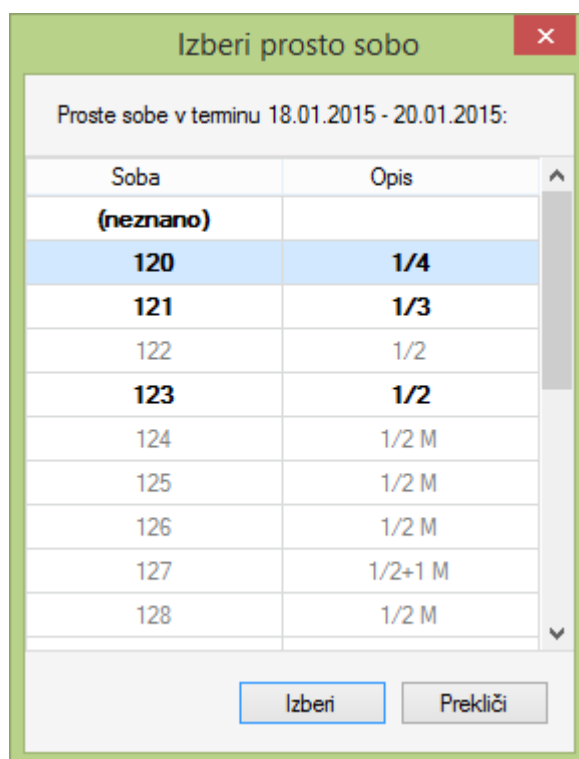
Ime rezervacije: NOVAK Barva: Zapiski: Cena: 90,00 € / noč + TTX
Št. oseb: 3 Prihod: 06.11.2014 Št. noči: 1 Če ostane 3 noči: -5% popust
Odhod: 07.11.2014
Zahteve: - Otroška postelja
- Parking
Soba: 121 Fiksna: ☐
OPC: ☐ do: 05.01.2015
Plačano: ☐ do: 07.11.2014

Potrdi vnos Zapri

Slika 4.24: Primer izpolnjenega obrazca za vpis nove rezervacije na ime *Novak* od *06. 11. 2014* do *07. 11. 2014*. Gost je ob rezervaciji najavil prihod 3 oseb, pri čemer je ena izmed njih novorojenček in potrebuje *otroško posteljo*. Gost je zahteval tudi *parkirno mesto* in se ob rezervaciji dogovoril za drugačno *ceno*, kot bi veljala po ceniku. Nastanitev je gostu obljubila *5% popust*, če rezervacijo podaljša na 3 *noči*. Uporabnik je zaradi posebnih zahtev gosta rezervacijo označil z *rumeno barvo*.



Slika 4.25: Prikazovalnik konfliktov omogoča hitro preurejanje dodeljenih sob v trenutku dodajanja nove rezervacije



Slika 4.26: Seznam za izbiro enote preveri, katere so v izbranem terminu na voljo, in jih poudari

kliku na polje izbire s seznama prostih enot v izbranem terminu (slika 4.26). Seznam vsebuje tudi enote, ki niso na voljo, in dovoli njihovo izbiro, vendar je uporabnik v takem primeru prisiljen razrešiti konflikte.

Ta funkcionalnost je zelo uporabna tudi v primeru, kadar uporabnik ne želi dodati nove rezervacije, ampak samo preveriti, katere enote so v določenem obdobju razpoložljive.

Urejanje

Kot je bilo že omenjeno, je mogoče termin rezervacije in dodeljeno sobo urediti tako, da se rezervacijo v mreži premakne na drugo mesto. Za urejanje drugih podrobnosti lahko uporabnik na rezervacijo dvakrat klikne ali pa jo s klikom označi in klikne na ustrezen gumb.

Urejanje rezervacije #803

Pregled in urejanje rezervacije:

Ime rezervacije: ZAJC Barva: Zapiski: Plačano z rač.: 3298 = 372,12 €

Št. oseb: 3 Prihod: 14.01.2015 Št. noči: 4

Odhod: 18.01.2015

Zahteve: - Zajtrk
- Parking

Soba: 121 Fiksna: ☐

OPC: ☐ do: 03.01.2015

Plačano: ☒ do: 18.01.2015

Gostje... Shrani in zapri Zapri

Slika 4.27: Obrazec za urejanje rezervacije (z rdečim kvadratom je označen gumb, ki omogoča vpisovanje gostov neposredno z obrazca)

Obrazec za urejanje rezervacije (na sliki 4.27) je vizualno enak obrazcu za dodajanje rezervacije, vendar vključuje dodatne kontrole:

- gumb za prikaz preglednega poročila rezervacije,
- gumb za stornacijo rezervacije ter
- gumb za neposredno vpisovanje podatkov gostov.

Hiro vpisovanje gostov

Uporabniku, ki uporablja urejevalnik rezervacij in želi vpisati podatke gostov za izbrano rezervacijo, ni potrebno to storiti na način, ki je opisan v razdelku 4.4.2. Pri obstoju rezervacije so nekateri podatki namreč že znani (kot na primer datuma prihoda in odhoda ter dodeljena enota), zato lahko uporabnik vpisovanje izvrši hitreje z uporabo namenskega gumba pri urejanju rezervacije.

Klik na gumb prikaže seznam podatkov vseh že vpisanih gostov in ponudi urejanje le-teh ter dodajanje novih. Pri dodajanju novih se polja za vpis podatkov, ki so že znani, samodejno prednastavijo.

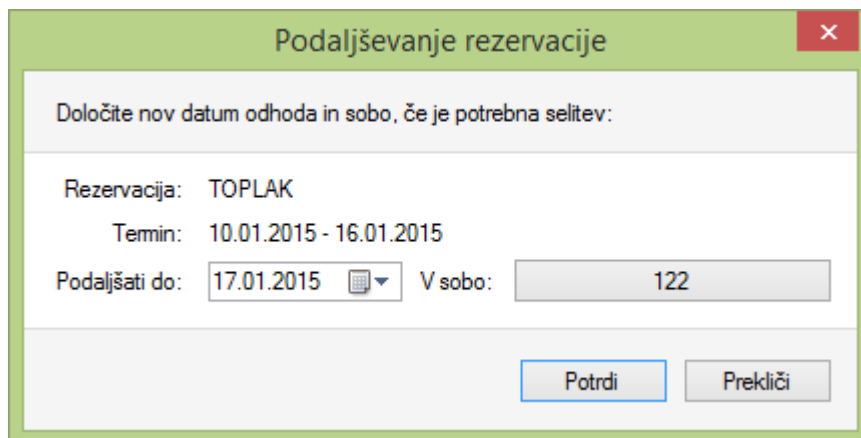
Samodejno upravljanje s knjigo gostov

Zelo pomembna funkcionalnost je ta, da ob spreminjanju rezervacije, za katero so bili podatki gostov že vpisani, program sam ugotovi, ali je potrebno urediti tudi vpise v knjigi gostov. Tukaj je vidna šibka sklopljenost, o kateri je bilo že veliko napisanega. Če vpisi gostov sovpadajo z rezervacijo, ki jo uporabnik spreminja (datumi odhodov in prihodov se ujemajo), vpise tiho popravi v ozadju. Če kakšen izmed vpisov ne sovпада z rezervacijo, je uporabnik o tem obveščen in mora izbrati, ali naj program popravi tudi ta vpis ali ne. V primeru, da vpis predstavlja gosta, ki je odšel prej od ostalih, je torej mogoče samodejno urejanje preskočiti.

Samodejno upravljanje s podatkovno bazo se ne izvaja samo ob njenem spreminjanju, ampak tudi ob operacijah podaljševanja in deljenja, ki sta opisani v nadaljevanju.

Podaljševanje bivanja

Omenjeno je že bilo, da gostje pogosto spreminjajo svoje rezervacije, še posebej kadar so že nameščeni v nastanitvi. Če želi gost rezervacijo skrajšati, je potrebno v urejevalniku rezervacij datum odhoda popraviti, kar ne more povzročiti težav.



Podaljševanje rezervacije

Določite nov datum odhoda in sobo, če je potrebna selitev:

Rezervacija: TOPLAK

Termin: 10.01.2015 - 16.01.2015

Podaljšati do: 17.01.2015 V sobo: 122

Potrdi Prekliči

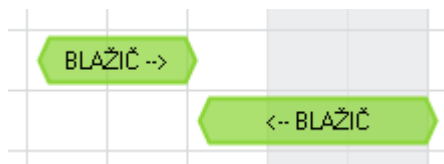
Slika 4.28: Orodje za podaljševanje rezervacije omogoča izbiro sobe, v katero je potrebno goste seliti

V primeru, da želi gost rezervacijo podaljšati, se lahko zgodi, da ne more ostati v isti enoti (na primer če za enoto že obstaja druga fiksna rezervacija). V tem primeru lahko uporabnik koristi namensko orodje za podaljševanje rezervacij, katerega vmesnik je viden na sliki 4.28.

Implementacija orodja lahko bistveno pohitri opisan proces, saj dobi uporabnik ob izbiranju nove enote informacijo, katere izmed njih so razpoložljive in ali taka sploh obstaja. Prav tako lahko poskusi z reševanjem konfliktov goste seliti v sobo, ki trenutno ni na voljo.

Delitev rezervacije

Zgodi se lahko, da gostov, ki so nastanjeni, ni potrebno seliti v drugo enoto, pa ti to vseeno zahtevajo (morda jim le-ta ni všeč ali želijo svojo rezervacijo nadgraditi z boljšo enoto). V tem primeru si lahko uporabnik pomaga z orodjem za delitev rezervacij, ki omogoča razbijanje rezervacij na več manjših samostojnih delov. V primeru selitve gostov na zahtevo lahko tako uporabnik rezervacijo razdeli na dva dela, pri čemer enega izmed njih premakne, kot je to prikazano na sliki 4.29.



Slika 4.29: Orodje za delitev razdeli rezervacijo na dve novi, pri čemer puščici ob imenih nakazujeta, da gre v resnici za eno samo

Skupinsko urejanje

Mreža omogoča sočasno izbiro večjega števila rezervacij s pridržanjem tipke *Shift* in klikanjem po rezervacijah. Za tako izbrano skupino rezervacij je nato mogoče izvesti nekatere skupinske operacije:

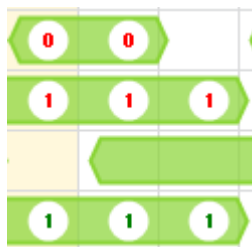
- izbira skupne barve izbranih rezervacij,
- fiksiranje vseh izbranih rezervacij,
- potrjevanje vseh nepotrjenih rezervacij v izboru,
- storniranje vseh izbranih rezervacij ter
- izdelava skupnega preglednega poročila.

Pogledi

Cilj implementacije različnih pogledov je ta, da lahko uporabnik že iz mreže vizualno ugotovi nekatera zanj pomembna dejstva. Prvi tak pogled, ki ga mreža omogoča, je prikaz števila vpisanih gostov (slika 4.30).

Pri kreiranju nove rezervacije mora uporabnik namreč določiti število najavljenih gostov, ki jih kasneje (ob njihovem prihodu) tudi vpiše. Z uporabo pogleda lahko uporabnik hitro ugotovi, če kateri izmed gostov ni bil vpisan ali če je v kako enoto vpisanih več gostov, kot jih je bilo najavljenih.

Drugi uporaben pogled je prikaz plačanih in neplačanih nočitev, ki ga lahko bralec vidi na sliki 4.31. Lahko se zgodi, da gost ob prihodu plača



Slika 4.30: Zelena števila pri prikazu števil vpisanih gostov uporabniku sporočajo, da je v knjigo gostov vpisanih toliko gostov, kolikor jih je najavljenih, rdeča pa, da se števili ne ujemata in je potreben pregled



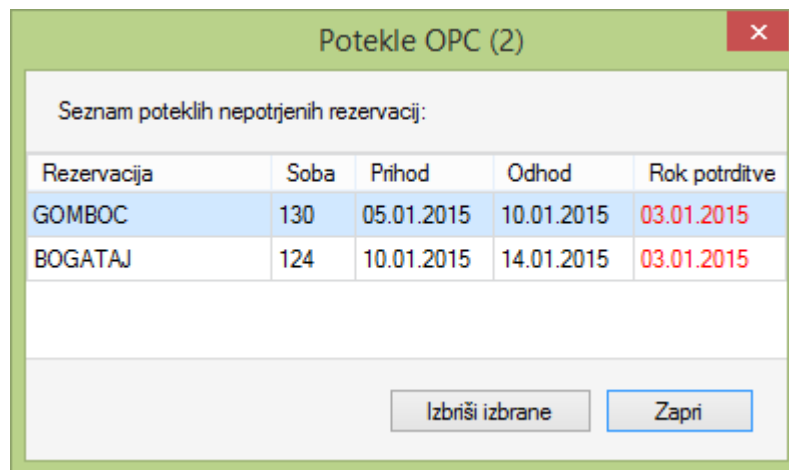
Slika 4.31: Zelene kljukice pri prikazu plačanih in neplačanih nočitev sporočajo, da je bila nočitev plačana, rdeči križci pa nasprotno

na primer 5 nočitev in kasneje rezervacijo podaljša. Nepozoren uporabnik bi lahko ob odhodu gosta spregledal, da je ostala ena nočitev neporavnana. Pogled tako omogoča hitro identifikacijo takih primerov.

Potekle nepotrjene rezervacije

Nepotrjenim rezervacijam prej ali slej poteče rok za potrditev. Če bi program take rezervacije samodejno brisal, bi to bilo za uporabnika zelo slabo, saj se lahko zgodi, da je gost na potrditev le pozabil ali pa da morda želi uporabnik v mreži hraniti tudi potekle nepotrjene rezervacije in jih izbrisati šele, ko je to nujno potrebno.

Zaradi tega program nepotrjene rezervacije v mreži obarva s sivo barvo, pretečenim pa ime obarva z rdečo barvo, da jih je lažje opaziti. Uporabniku, ki želi razpoložljivost povečati z brisanjem vseh ali nekaterih pretečenih nepotrjenih rezervacij, je na voljo namensko orodje (slika 4.32), ki poišče vse



Seznam poteklih nepotrjenih rezervacij:				
Rezervacija	Soba	Prihod	Odhod	Rok potrditve
GOMBOC	130	05.01.2015	10.01.2015	03.01.2015
BOGATAJ	124	10.01.2015	14.01.2015	03.01.2015

Izbriši izbrane Zapri

Slika 4.32: Primer rezultata iskanja vseh poteklih nepotrjenih rezervacij

pretečene rezervacije in nudi njihovo urejanje ali hitro brisanje.

Poročila

Uporaba urejevalnika rezervacij poleg knjige gostov pomeni, da program vsebuje tudi podatke o nočitvah, ki še niso bile izvedene. V tem primeru je mogoče izdelati bogatejše poročilo dnevnega stanja enot od tistega, ki ga kreira knjiga gostov, saj je možno v njem navesti:

- v katere enote gostje prihajajo ter
- koliko noči so ostale enote proste (kar je razlika v dnevih med trenutnim datumom in datumom prihoda prve rezervacije, ki za enoto obstaja).

Poročilo lahko pride zelo prav, še posebej v recepcijah nastanitev, ki imajo nekoliko večje število enot. V primeru prihoda nenajavljenega gosta ta izrazi, koliko noči želi prenočevati, in uporabnik lahko iz poročila hitro ugotovi, v katero enoto bi ga bilo najboljše namestiti, da ostane razpoložljivost optimalna. Primer izrezka je prikazan na sliki 4.33.

Drugo poročilo, ki ga je mogoče kreirati z urejevalnikom rezervacij in lahko uporabniku izjemno olajša delo, je poročilo o razpoložljivosti enot po tipih za

	120	1/4	PRIHOD: JUG -->
	124	1/2 M	ZASEDENO ČUK
	125	1/2 M	ODHOD: BOGATAJ PROSTO 2 NOČI
	126	1/2 M	ODHOD: PRIMOŽIČ PRIHOD: MAČEK

Slika 4.33: Primer izreza poročila o dnevnem stanju enot, ki ga kreira urejevalnik rezervacij

izbran termin. Primer je prikazan na sliki 4.34, motivacija za implementacijo pa je bila podana že v podpoglavju 2.1.5.

4.4.3 Dodatna orodja

Poleg knjige gostov in urejevalnika rezervacij sta implementirani tudi dve dodatni preprosti orodji, ki za implementacijo nista posebej zahtevni, a lahko uporabniško izkušnjo izboljšata.

Prvo izmed teh je preprost urejevalnik besedila, s katerim lahko uporabnik hitro oblikuje dokument, ki ga želi shraniti, poslati ali natisniti. Če želi, lahko oblikuje tudi predloge, ki jih po potrebi uporabi za lastne potrebe.

Drugo, morda pomembnejše, je integriran spletni brskalnik. Uporabnik ima lahko ob programu Opengost odprte tudi druge programe, vendar je delo lahko prijetnejše, če so vsi pripomočki, ki jih pogosto uporablja, na enem mestu. S pridrženjem tipke *Shift* in klikom na gumb v glavnem meniju lahko uporabnik uporabniški vmesnik razdeli na dva dela, pri čemer vsak vsebuje drug modul. Primer je prikazan na sliki 4.35.

	05.01.	06.01.	07.01.	08.01.	09.01.	10.01.	11.01.	12.01.	13.01.	14.01.	15.01.	16.01.	17.01.	18.01.
	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)
1/2	6 (3)	5 (4)	2 (7)	3 (6)	2 (7)	2 (7)	1 (8)	3 (6)	3 (6)	2 (7)	3 (6)	2 (7)	0 (9)	2 (7)
1/2 M	2 (5)	2 (5)	1 (6)	1 (6)	2 (5)	1 (6)	1 (6)	3 (4)	3 (4)	2 (5)	4 (3)	2 (5)	0 (7)	3 (4)
1/2+1 M	0 (2)	0 (2)	0 (2)	2 (0)	2 (0)	0 (2)	0 (2)	0 (2)	0 (2)	1 (1)	1 (1)	2 (0)	1 (1)	2 (0)
1/3	1 (0)	0 (1)	0 (1)	0 (1)	0 (1)	0 (1)	0 (1)	1 (0)	1 (0)	0 (1)	0 (1)	0 (1)	0 (1)	1 (0)
1/4	1 (1)	1 (1)	1 (1)	1 (1)	1 (1)	1 (1)	2 (0)	0 (2)	0 (2)	1 (1)	0 (2)	0 (2)	0 (2)	2 (0)

Slika 4.34: Primer poročila o razpoložljivosti sob po tipih za izbran termin (z besedilom sta podana podatka o številu prostih in številu zasedenih enot pripadajočega tipa, z intenziteto rdeče bave pa je izražena zasedenost)

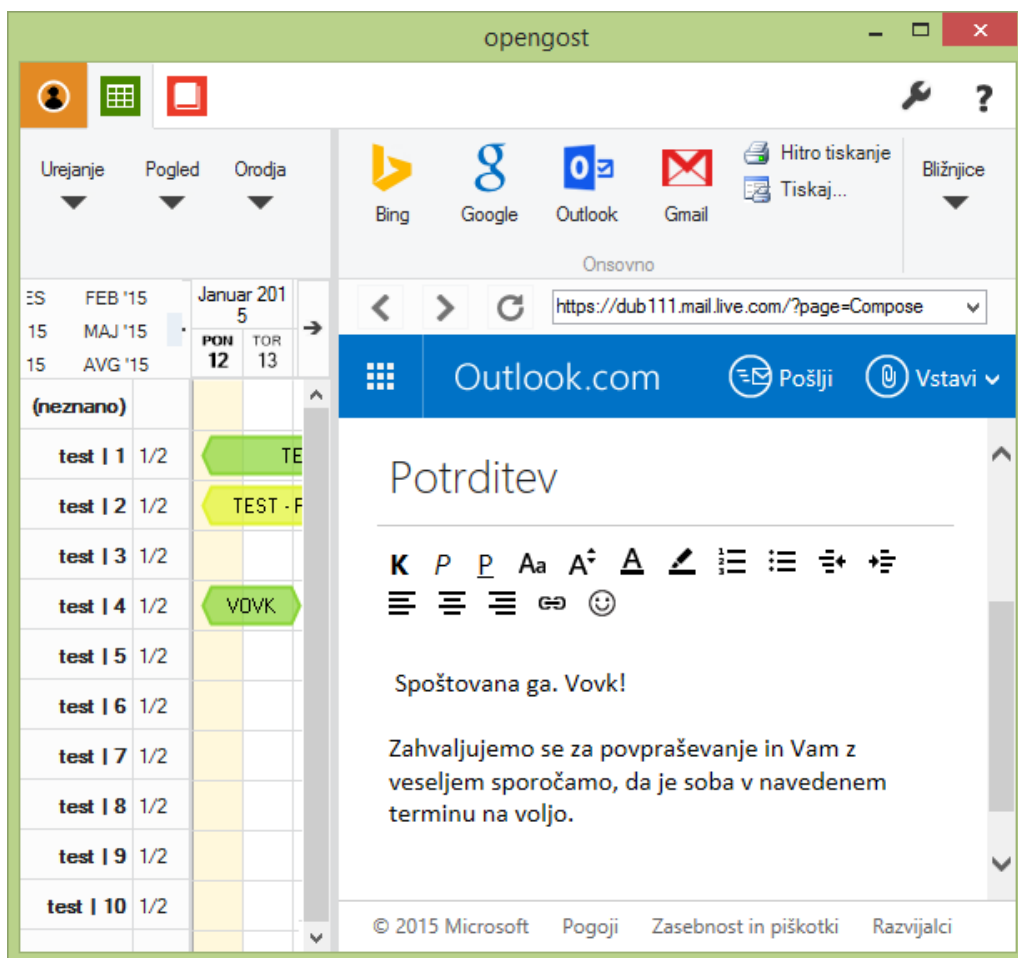
4.4.4 Namestitveni program

Razvijalci komercialne programske opreme se redko potrudijo, da bi bil namestitveni program njihove rešitve enostaven za uporabo. Poleg svoje programske opreme namreč pogosto nudijo tudi namestitev, ki jo nekateri dodatno zaračunajo, zato jim implementacija takega namestitvenega programa, ki bi omogočal uporabniku, da namestitev opravi sam, ni v interesu.

Rešitev Opengost kot brezplačna rešitev brez fizične servisne podpore potrebuje dober in zanesljiv namestitveni program, ki samodejno namesti vsa potrebna orodja (sistem za upravljanje podatkovne baze ter ogrodje .NET 4.5) in zna program samodejno posodobiti. Implementacija namestitvenega programa ni enostavno opravilo in zahteva veliko časa.

Z namenom razbremenitve razvijalcev je Microsoft v ogrodje .NET vključil tudi možnost uporabe samodejno generiranega namestitvenega programa, ki sloni na lastni tehnologiji ClickOnce. Tako izdelan namestitveni program je enostaven za uporabo, poleg tega pa:

- samodejno namesti vsa potrebna orodja, ki jih program potrebuje za delo,



Slika 4.35: Primer sočasne uporabe urejevalnika rezervacij in integriranega brskalnika, v katerem uporabnik uporablja storitev Microsoft Outlook za pošiljanje elektronske pošte

- samodejno preveri, ali je na voljo posodobitev programa, in posodobitev tudi namesti ter
- omogoča obnavljanje programa na starejšo različico, če uporabniku posodobitev ne odgovarja.

Samodejno preverjanje in nameščanje posodobitev je pri iteracijsko inkrementalnem razvoju zelo dobrodošlo, saj se program neprestano spreminja. Če razvijalec proces posodabljanja prepusti uporabniku, obstaja velika verjetnost, da posodobitev, če je trenutno zadovoljen s programom, niti ne bo namestil. S tem se uporabnik izloči iz razvojnega procesa, česar razvijalec ne želi.








Opisan način izdelave namestitvenega programa deluje kot nalašč za potrebe lastne rešitve, zato je bil tudi uporabljen.

4.4.5 Spletna stran s forumom

Za predstavitev lastne rešitve ciljnim uporabnikom je bila ustvarjena preprosta spletna stran [17], ki jo sestavljajo:

- opis programa in njegovih funkcionalnosti,
- stran za prenos namestitvenega programa,
- bogat uporabniški priročnik in
- uporabniški forum.

Motivacija za implementacijo uporabniškega foruma (slika 4.36) je bila podana že v opisu iterativno inkrementalnega razvoja v podpoglavju 4.1.1.

<input type="checkbox"/>		Pogosta vprašanja in odgovori (1) Avtor: jaz - 1 objava - 26 ogledov 	6. jan.
<input type="checkbox"/>		Novice, nove različice in splošne objave Avtor: jaz - 2 objavi - 29 ogledov 	6. jan.
<input type="checkbox"/>		Predstavitev uporabnikov (1) Avtor: jaz - 1 objava - 24 ogledov 	5. jan.
<input type="checkbox"/>		Predlogi dopolnitev Avtor: miro....@con-dar.com - 2 objavi - 16 ogledov	8. jan.

Slika 4.36: Na uporabniškem forumu se je že tri dni po objavi spletne strani pojavilo nekaj komentarjev in predlogov za naslednjo iteracijo razvoja

Poglavje 5

Odziv prvih uporabnikov

Lastna rešitev in spletna stran sta bili objavljeni 05. januarja 2015, naslednjega dne pa je bilo nekaterim potencialnim uporabnikom odposlano elektronsko sporočilo s predstavitvijo programa.

Že dan po objavi je bilo prejetih dobrih deset odgovorov, v katerih so se pošiljatelji zahvalili za ponujeno rešitev, saj so bodisi svoje goste doslej prijavljali brez programske opreme ali pa z uporabljenimi rešitvijo niso bili zadovoljni. Med njimi je bilo nekaj takih, v katerih so pošiljatelji razkrili, da so že dalj časa iskali rešitev, ki bi bila predvsem enostavnejša za uporabo in cenejša. Tisti, ki so program tudi takoj namestili in preizkusili, so podali zelo pozitivna mnenja. Predvsem sta jih navdušila uporabniški vmesnik in enostavnost namestitve.

V naslednjih dneh je bilo število odposlanih elektronskih sporočil postopoma zmeraj večje. Z objavo spletne strani v spletnih iskalnikih (kot so Google, Bing, Najdi.si in podobni) in na socialnih omrežjih (Facebook in Google+) je nekaj uporabnikov o programu izvedelo brez elektronskega sporočila. Postopoma sta pričeli rasti tudi število odgovorov in obiskanost spletne strani.

Odziv prvih uporabnikov je bil zelo spodbuden. Veliko jih je v odgovoru zapisalo, da bodo do konca meseca še uporabljali dosedanjo rešitev, z novim mesecem pa to zamenjali s programom Opengost.

Ob tem pa je bilo v nekaterih odgovorih opaziti precejšnjo mero skeptičnosti, saj je uporabnike pogosto zanimalo, koliko časa bodo lahko program *za-res* uporabljali brezplačno, koliko bodo prisiljeni plačevati za vzdrževanje ter kakšne bodo omejitve funkcionalnosti za brezplačne uporabnike¹. Nivo skeptičnosti lepo nakazuje na odnos med uporabniki in trenutnimi rešitvami, na katere uporabniki gledajo kot na nekaj, kar *morajo* uporabljati, ne pa na nekaj, kar bi si uporabljati *želeli*.

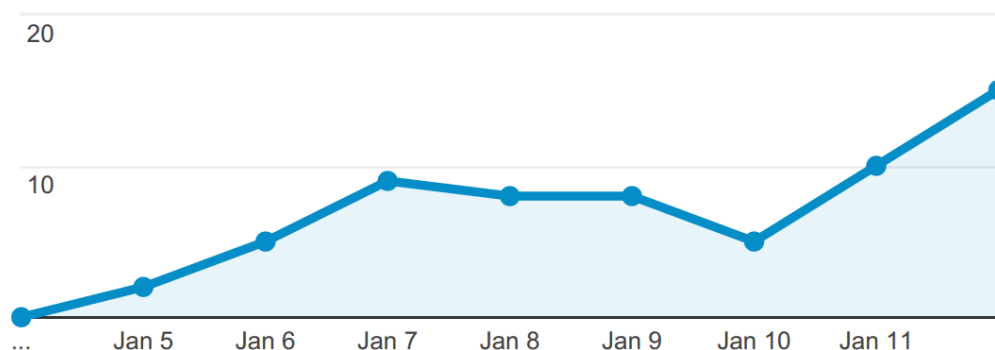
5.1 Analitika

Za ugotavljanje obiskanosti spletne strani in števila uporabnikov aplikacije je uporabljeno zmogljivo brezplačno orodje Google Analytics. Spremljanje števila uporabnikov in obiskovalcev je za razvijalca programske opreme ključnega pomena, saj lahko le na ta način z veliko zanesljivostjo med drugim ugotavlja [18]:

- ali potencialne uporabnike programska oprema zanima,
- kolikšen delež uporabnikov se za uporabo ponujene rešitve odloči,
- katere odločitve v razvoju so dobre (opazno po povečanjem številu uporabnikov) in katere slabe (število uporabnikov se zmanjša) ter
- kakšna je dinamika uporabe.

Spremljanje števila aktivnih uporabnikov programa je implementirano tako, da program med tekom v rednem intervalu preko spleta (če je internetna povezava na voljo) pošlje anonimni signal na strežnik storitve Google Analytics. Signali zavoljo ohranjanja zasebnosti ne nosijo nobene informacije, razen unikatne naključno generirane številke namestitve, ki je potrebna, da se lahko loči med signali različnih uporabnikov.

¹Na spletni strani je sicer zapisano, da je program popolnoma brezplačen in da bo tak tudi ostal, vendar so uporabniki očitno vajeni takih trditev in obljub, ki se jih razvijalci preredko držijo



Slika 5.1: Število unikatnih uporabnikov programa Opengost po dnevih v prvem tednu od objave

Sliki 5.1 in 5.2 prikazujeta rezultate, pridobljene v prvem tednu od objave rešitve in spletne strani.

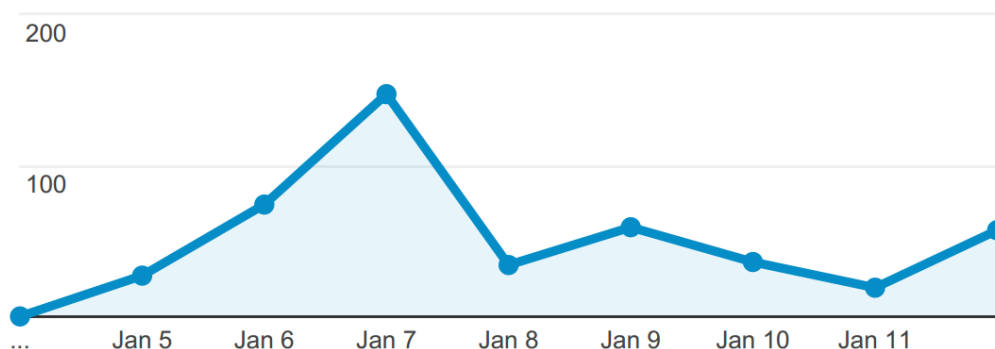
Večjih sklepov iz tako kratkega opazovalnega obdobja ni mogoče tvoriti, opaziti pa je mogoče naslednje:

- potencialnih uporabnikov, ki jih podrobnosti o rešitvi zanimajo, je veliko (dva dni po objavi je spletno stran obiskalo kar 147 različnih obiskovalcev, kar je za popolnoma novo, neznano spletno stran veliko) ter
- število uporabnikov, ki si program namesti in ga nato tudi uporablja, narašča (v zadnjem dnevu prvega tedna od objave je število različnih uporabnikov, ki so že pričeli z uporabo programa, naraslo na 15).

5.2 Tržni potencial in poslovni model

Ob vsem delu in trudu, ki ga zahteva neprekinjen razvoj lastne rešitve, se je smiselno vprašati, ali ima program kljub brezplačnosti tržni potencial.

Omenjeno je že bilo, da je lastna rešitev delno zgrajena iz odprtokodnih komponent, zato je potrebno preučiti, kakšne so posledice njihove uporabe. Odprtokodne komponente so namreč lahko ponujene pod različnimi licencami, ki njihovo uporabo omejujejo na različne načine. Nekatere so take,



Slika 5.2: Število unikatnih obiskovalcev spletne strani po dnevih v prvem tednu od objave

da je redistribucija komponent dovoljena, druge pa zahtevajo, da jih končni uporabnik pridobi pri njihovih avtorjih. Licence, ki redistribucijo dovoljujejo, pa lahko od razvijalca, ki jih uporablja v svoji rešitvi, zahtevajo, da tudi svojo rešitev ponudi brezplačno in njeno kodo opremi z enako licenco. Najmanj stroge oblike licenc razvijalca z vidikov redistribucije in plačljivosti ne omejujejo. Primer take vrste licenciranja predstavljata licenca MIT (*Massachusetts Institute of Technology License*) in LGPL (*GNU Lesser General Public License*).

Pri implementaciji programa Opengost so bile uporabljene le take odprtokodne komponente, ki jih avtorji ponujajo pod eno izmed omenjenih nestrogih licenc. Brezplačnost programa tako ni zahtevana, avtorju pa izvirne kode ni potrebno objaviti. Uporaba odprtokodnih komponent tako ne zmanjšuje tržnega potenciala lastne rešitve.

Motivacija, da se program Opengost ponudi brezplačno, izvira iz inovativnega poslovnega modela Freemium, ki se je izkazal za zelo uspešnega [19]. Začetek takega poslovnega modela predstavlja produkt, ki se uporabnikom ponudi brezplačno. Produkt ima zaradi tega veliko konkurenčno prednost in zagotavlja hitrejše pridobivanje uporabnikov. Če je produkt kvaliteten, se s tem prične graditi skupina zadovoljnih uporabnikov, ki razvijalcu zaupajo in se na produkt navadijo. Razvijalec svojo rešitev razvija dalje in s tem kaže,

da ima produkt prihodnost.

Ko je skupina uporabnikov brezplačnega produkta dovolj velika, razvijalec svoji rešitvi doda napredne funkcionalnosti, za katere zahteva doplačilo. Ob tem brezplačne različice ne sme omejiti, saj bi bili s tem uporabniki ogoljufani. Prav tako ne sme prekiniti njenega razvoja, saj s tem izgubi zaupanje. Plačljive dodatne funkcionalnosti tako predstavljajo le nadgradnjo, za katero se lahko uporabnik odloči ali ne. Če se zanjo ne odloči, njegova dotedanja uporabniška izkušnja ostane nespremenjena.

Ker je uporabnikov brezplačne različice običajno razmeroma veliko, ti pa razvijalcu zaupajo, je tako možnost, da se le-ti odločijo za doplačilo, velika.

Brezplačno različico programa Opengost predstavljata elektronska knjiga gostov in urejevalnik rezervacij. Ker sta ta modula manjšim nastanitvenim objektom najnujnejša, je mogoče pričakovati, da se bo razmeroma veliko uporabnikov odločilo za uvedbo ali prehod na rešitev Opengost. Z nadaljnjim razvojem in vključevanjem uporabnikov vanj se bo skupina uporabnikov povečevala. V primernem trenutku bo tako dobro premisliti o razvoju drugih modulov, ki jih Opengost še ne vključuje (na primer upravljalca kanalov in blagajne). V primeru zadostnega interesa jih je smiselno razviti in ponuditi kot doplačljive nadgradnje, pri čemer se bo zagotovo del uporabnikov odločil zanje.

5.3 Nadaljnji razvoj

Trditev, da je implementacija uporabniškega foruma učinkovita metoda vključevanja uporabnikov v razvoj, se je izkazala za resnično. Nekaj dni po objavi lastne rešitve je bilo na uporabniškem forumu že objavljenih nekaj predlogov za nadaljnji razvoj. Množica teh pa je prispela tudi v obliki e-pošte.

Zanimivo je pogledati nekaj izmed njih:

- večja izbira barv za označevanje rezervacij,
- možnost vnosa vira rezervacije in pregled rezervacij po viru v izbranem obdobju,

- uporabnika naj program opozori, če poskusi rezervacijo premakniti v sobo, ki je drugačnega tipa od sobe, ki ji je trenutno dodeljena, in
- dodati možnost samodejnega izdelovanja varnostne kopije podatkov (program omogoča izdelavo varnostne kopije, vendar uporabniki na to pogosto pozabijo, zato bi ta možnost prišla prav).

Bralec lahko opazi, da pri naštetih predlogih ne gre za velike spremembe, pač pa za malenkosti, ki bi se utegnile marsikateremu razvijalcu zdeti nepotrebne. A dejstvo, da so predlogi bili sporočeni v tako kratkem času, nakazuje, da uporabniki (morda ne vsi, vendar vsaj nekateri med njimi) omenjene funkcionalnosti močno pogrešajo. Iz zadnjih dveh predlogov bi bilo mogoče sklepati, da je uporabnik že doživel slabo izkušnjo pri drugi rešitvi in da si želi, da bi rešitev, ki jo uporablja, ponovitev slabe izkušnje preprečila.

Sporočene predloge je tako v naslednji iteraciji razvoja potrebno implementirati ne glede na to, kako pomembni se razvijalcu zdijo. Pričakovati je, da bo s porastom števila uporabnikov tudi predlogov zmeraj več, zato bo manjših razvojnih iteracij še veliko.

Poglavje 6

Sklepne ugotovitve

Bralec je v besedilu recepcijsko delo v manjših nastanitvenih objektih dobro spoznal in najverjetneje med branjem tudi sam dobil idejo ali dve, kako bi ga bilo mogoče z uporabo programske opreme olajšati, pohitriti in izboljšati. Če se je pri tem tudi vprašal, kako bi svoje ideje implementiral na način, da bi bile za uporabnika samoumevne in da ga pri delu ne bi omejevale, je že na dobri poti k zasnovi dobre rešitve.

V tretjem poglavju je bilo pokazano, da je obstoječih rešitev za manjše nastanitve malo in da imajo te kar veliko pomanjkljivosti - nekatere v preprečevanju napak, druge v fleksibilnosti, ostale pa v uporabniški izkušnji, ki jo povzroča nepregleden in zapleten uporabniški vmesnik.

Četrto poglavje je bilo tako namenjeno zasnovi in implementaciji lastne rešitve, ki poskuša pomanjkljivosti odpraviti. Podana so bila načela, ki se jih mora ob tem razvijalec držati. Iterativno inkrementalni in uporabniško usmerjen razvoj je odličen pristop k razvoju programa, za katerega razvijalec ne more natanko vedeti, ali bo uporabniku zares olajšal delo ali ne. Medtem ko trislojna arhitektura od razvijalca zahteva nekaj več dela in navora, je ta za preživetje rešitve v tako dinamičnem razvojnem modelu bistvenega pomena. Ob vsem tem pa je veliko bolje, če razvijalec posveti več časa izpopolnjevanju funkcionalnosti enega samega modula, kot pa hitri implementaciji čimvečjega števila le-teh.

Implementacija lastne rešitve v obliki programa Opengost in njegove funkcionalnosti je bila podrobno opisana in lahko bralcu, ki želi razviti ali izpolniti svojo lastno rešitev, služi kot izčrpen seznam na videz nepomembnih izboljšav, ki uporabnikom v praksi veliko pomenijo.

Rezultati spremljanja števila obiskovalcev spletne strani in uporabnikov programa v prvem tednu ne nudijo dovolj informacij, da bi bilo mogoče brez dvoma sklepati o tem, ali je bil namen lastne rešitve popolnoma dosežen ali ne, vendar rastoč trend števila uporabnikov, njihovi pozitivni prvi komentarji in hitro sodelovanje pri razvoju nakazujejo, da se lahko programu Opengost in skupnosti njegovih uporabnikov obeta lepa prihodnost.

Literatura

- [1] Vlada Republike Slovenije, “Strategija razvoja slovenskega turizma 2012–2016.” Dostopno na:
http://www.mgrt.gov.si/fileadmin/mgrt.gov.si/pageuploads/turizem/Turizem-strategije_politike/Strategija_turizem_sprejeto_7.6.2012.pdf, 2012.
- [2] Statistični urad Republike Slovenije, “Prenočitvene zmogljivosti po vrstah občin in po vrstah nastanitvenih objektov, slovenija, letno,” 2013.
- [3] J. A. Bardi, *Hotel front office management*. John Wiley & Sons, Inc., 1996.
- [4] B. Kranjc, “Razvoj programske opreme za vodenje turističnih nastanitev (diplomsko delo).” Fakulteta za računalništvo in informatiko, Univerza v Ljubljani, 2009.
- [5] W. O. Galitz, *The essential guide to user interface design: an introduction to GUI design principles and techniques*. John Wiley & Sons, 2007.
- [6] C. Larman and V. R. Basili, “Iterative and incremental development: A brief history,” *IEEE Computer*, vol. 36, no. 6, pp. 47–56, 2003.
- [7] V. R. Basili and A. J. Turner, “Iterative enhancement: A practical technique for software development,” *Software Engineering, IEEE Transactions on*, no. 4, pp. 390–396, 1975.

- [8] J. M. Greenbaum and M. Kyng, *Design at work: Cooperative design of computer systems*. L. Erlbaum Associates Inc., 1991.
- [9] J. Hess, S. Offenbergh, and V. Pipek, "Community driven development as participation?: involving user communities in a software design process," in *Proceedings of the Tenth Anniversary Conference on Participatory Design 2008*, pp. 31–40, Indiana University, 2008.
- [10] W. W. Eckerson, "Three tier client/server architectures: achieving scalability, performance, and efficiency in client/server applications," *Open Information Systems*, vol. 3, no. 20, pp. 46–50, 1995.
- [11] A. Troelsen, *Pro C# 5.0 and the .NET 4.5 Framework*. Apress, 2012.
- [12] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowledge acquisition*, vol. 5, no. 2, pp. 199–220, 1993.
- [13] J. Martin, "Managing the data base environment," 1981.
- [14] T. Berners-Lee, J. Hendler, O. Lassila, *et al.*, "The semantic web," *Scientific american*, vol. 284, no. 5, pp. 28–37, 2001.
- [15] M. Hepp, "Accommodation ontology language reference," 2013.
- [16] D. A. Norman, *The design of everyday things*. Basic books, 2002.
- [17] A. Bešir, "Predstavitevna spletna stran programa opengost." Dostopno na: <http://www.opengost.si/>, 2015.
- [18] B. Clifton, *Advanced web metrics with Google Analytics*. John Wiley & Sons, 2012.
- [19] M. F. Niculescu and D. J. Wu, "When should software firms commercialize new products via freemium business models?," *Workshop on Information Systems and Economics*, 2011.